— VISUAL OBJECT SEGMENTATION —

BASED ON TEMPORAL AND LINGUISTIC CUES

ZHAO YANG

CHRIST CHURCH



Department of Engineering Science
University of Oxford

A dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy

Trinity 2022

ADVISOR

PROFESSOR PHILIP H.S. TORR

To the past and future.

ACKNOWLEDGEMENTS

I used to fear life's uncertainties but now I embrace them. This change of mindset happened gradually for me during this PhD program and it was induced by people with whom I've had the greatest fortune to work, some of the world's brightest and most capable individuals. I'd like to use the space here to express my sincerest and deepest appreciation to them.

I'd like to thank my supervisor, Prof. Philip Torr, for guiding me to the world of scientific research and never wavering in his support to help me reach my potential. He provided guidance that was the most needed, resources that were the most valuable, understanding that was the most generous, and ideas that were the most ingenious. By setting an example of the highest standards himself, he inspired me to grow, as a professional, to acquire knowledge and research skills, and as a person, to build character in the face of adversity. Time spent in the Torr Vision Group and with my supervisor was immensely rewarding and truly broadened my horizons, and for that I'm deeply indebted to Prof. Torr.

I'd like to thank Dr. Luca Bertinetto, Dr. Qiang Wang, Prof. Song Bai, Prof. Yansong Tang, Prof. Hengshuang Zhao, and Prof. Li Zhang for having been such great friends and mentors—you taught me the fundamentals of research, and you showed me what contenders and winners are made of, qualities of which are among the countless things I try to emulate from you.

I'd like to thank Dr. Jiaqi Wang and Dr. Kai Chen at the Shanghai AI Lab for their great collaboration and invaluable friendship. I'd like to thank Prof. Devon Hjelm and Prof. Yoshua Bengio for their exceptional guidance and support during my project at Mila.

I am grateful to Prof. Andrea Vedaldi and Prof. Victor Prisacariu for being my assessors of Transfer of Status and Confirmation of Status, and for having provided insightful feedback that helped me stay on track. And I am very grateful to Prof. Andrea Vedaldi and Prof. Jiaya Jia for being the examiners of my viva voce.

I'd also like to extend my sincere thanks to Dr. Stuart Golodetz, Dr. Oscar Rahnama, and Prof. Kai Han for their advice on many things, and Ms. Joanna

Zapisek, Ms. Cassandra Warren, Mr. Adam Galt, Mr. Christopher Rabson, and Mr. Aenaes Ward for having taken care of me time and again.

Last but not least, I'd like to thank my parents for their love beyond measure.

DECLARATION

I hereby declare that this thesis is submitted to the Department of Engineering Science, University of Oxford, in partial fulfillment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

September 2022
Zhao Yang

ABSTRACT

The aspiration of computer vision is to enable computers to make sense of a world, one that exists in a vast sea of colorful pixels reflecting what humans see every day. Central to this goal perhaps, is the discovery of object representations from these pixels: clusters of pixels representing the things from big to small that constitute our physical world and the interactions with which form our daily life. Visual object segmentation is a subject that deals with this task. It concerns the prediction of pixel-wise masks that delineate objects of interest from visual data. Different types of evidence have been leveraged for segmenting out objects (according to the application), *e.g.*, contours, saliency, motion, extreme points, and pre-defined semantic categories, among others. In this thesis, we investigate the exploitation of temporal correspondences and natural language descriptions for visual object segmentation, motivated by their omnipresence and efficacy, and contribute new methods that improve the state of the art for the relevant applications.

In the first part of the thesis, we consider the modeling of temporal dependencies for object segmentation in videos, where the continuous movement of an object offers natural correspondences in time. Conventional methods based on sequential modeling are ineffective at exploiting such cues, and they still perform on a par with static segmentation models. In this thesis, we address this issue by introducing a technique that learns dense correspondences between pixels of frames that can be arbitrarily far apart in time. This enables the learning of long-term dependencies without conditioning on intermediate frames, which significantly improves the (deep) representation consistency of the object over time and leads to accurate object segmentation in videos.

In the second part, we explore the use of natural language for object segmentation in images, where an object that corresponds to a natural language description is segmented out. With the aim of aligning linguistic meanings and visual clues in a common feature space, previous methods adopt the paradigm of fusing visual and linguistic features after they have been independently extracted from a vision encoder and a language encoder, respectively. In this thesis,

we introduce a new paradigm that relocates cross-modal feature alignment to an earlier phase, *i.e.*, to be conducted jointly with image encoding. The main idea is to supplement linguistic features to multiple intermediate layers of a vision Transformer network, to have linguistic information embedded jointly with visual information by the Transformer layers during the forward pass. With this approach, we tap into the correlation modeling power of the Transformer model at an early stage, and can harvest segmentation with a lightweight mask predictor.

In the last part, we investigate the joint utilization of temporal and linguistic cues for visual object segmentation in the task of video object segmentation from referring expressions. We show how appearance, language, and motion features can be simultaneously aligned in a single network that is end-to-end trainable by exploiting the hierarchical structure of convolutional neural networks. Compared with previous methods, the proposed method has a much-enhanced capability in representing multi-modal inputs at different semantic and spatial granularity levels while enjoying a much simpler design.

CONTENTS

1 Introduction **1**
- 1.1 Visual segmentation . . . 2
- 1.2 Visual object segmentation . . . 4
  - 1.2.1 Automatic video object segmentation . . . 6
  - 1.2.2 Image segmentation from referring expressions . . . 8
  - 1.2.3 Video object segmentation from referring expressions . . . 9
- 1.3 Contributions and thesis outline . . . 11
- 1.4 Publications . . . 13

2 Literature Review **15**
- 2.1 Video object segmentation . . . 15
- 2.2 Referring image segmentation . . . 19
- 2.3 Video object segmentation from referring expressions . . . 26

3 Anchor Diffusion for Unsupervised Video Object Segmentation **32**
- 3.1 Introduction . . . 33
- 3.2 Related work . . . 36
- 3.3 Method . . . 38
  - 3.3.1 Anchor diffusion . . . 40
- 3.4 Experiments . . . 43
  - 3.4.1 Benchmarks . . . 45
  - 3.4.2 Ablation studies . . . 46
  - 3.4.3 Comparison with the state of the art . . . 47
- 3.5 Discussion . . . 52
- 3.6 Appendices . . . 56
  - 3.6.1 Global comparison . . . 56
  - 3.6.2 Per-sequence comparison . . . 56
  - 3.6.3 Qualitative analysis on FBMS and ViSal . . . 59
  - 3.6.4 Foreground correspondences analysis . . . 60
  - 3.6.5 Instance pruning . . . 62
- 3.7 Statement of authorship . . . 62

4 LAVT: Language-Aware Vision Transformer for Referring Image Segmentation 64
4.1 Abstract . . . 64
4.2 Introduction . . . 65
4.3 Related work . . . 67
4.4 Method . . . 70
4.4.1 Language-aware visual encoding . . . 70
4.4.2 Pixel-word attention module . . . 71
4.4.3 Language pathway . . . 74
4.4.4 Segmentation . . . 75
4.4.5 Implementation . . . 75
4.5 Experiments . . . 76
4.5.1 Datasets and metrics . . . 76
4.5.2 Comparison with others . . . 77
4.5.3 Ablation study . . . 78
4.6 Discussion . . . 84
4.7 Appendices . . . 86
4.7.1 Potential biases of the language model . . . 86
4.7.2 The language pathway . . . 86
4.7.3 Precision-recall analysis . . . 87
4.7.4 Mean IoU . . . 88
4.7.5 Visualizations . . . 88
4.8 Statement of authorship . . . 90

5 Hierarchical Interaction Network for Video Object Segmentation from Referring Expressions 91
5.1 Introduction . . . 92
5.2 Related work . . . 95
5.3 Method . . . 96
5.3.1 Multi-modal feature encoding . . . 97
5.3.2 Hierarchical feature interaction . . . 98
5.3.3 Multi-modal feature decoding . . . 100
5.4 Experiments . . . 100
5.4.1 Implementation details and evaluation metrics . . . 101
5.4.2 Comparison with others . . . 101
5.4.3 Ablation study . . . 104
5.5 Discussion . . . 107

5.6 Appendices 108
5.6.1 Implementation details 109
5.6.2 Datasets and evaluation metrics 110
5.6.3 Attribute-based performance on DAVIS-16 112
5.6.4 Ablative qualitative analysis 112
5.6.5 The effects of optical flow 114
5.7 Statement of authorship 125

6 Conclusion **126**
6.1 Future work 128

References **133**

LIST OF FIGURES

1.1 Examples of visual segmentation. 3
1.2 Three popular formulations for the problem of visual object segmentation. 5
1.3 Illustration of the task of automatic video object segmentation. . 7
1.4 Illustration of the task of image segmentation from referring expressions. 8
1.5 Illustration of the task of video object segmentation from referring expressions. 10

2.1 Illustration of the task of referring expression comprehension. . . 26
2.2 Illustration of the task of actor-action segmentation from a sentence. 27

3.1 One-to-many similarities learned by the proposed AD-Net. . . . 34
3.2 An overview of the AD-Net architecture. 39
3.3 A temporal analysis of embeddings learned by the AD-Net. . . . 43
3.4 Precision-recall curves of AD-Net on DAVIS, FBMS, and ViSal. . 51
3.5 Example predictions of the AD-Net on the DAVIS-16 validation set. 52
3.6 Mean region similarity of AD-Net on individual videos from DAVIS-16. 58
3.7 Mean contour accuracy of AD-Net on individual videos from DAVIS-16. 58
3.8 Example predictions of the AD-Net on challenging videos from FBMS. 59
3.9 Example predictions of the AD-Net on challenging videos from ViSal. 60
3.10 Illustrations of temporal pixel similarities learned by AD-Net. . . 61

4.1 A comparison between the proposed LAVT network and previous methods. 66
4.2 An overview of the proposed LAVT network architecture. 69

4.3 An illustration of the proposed pixel-word attention module. . . 72
4.4 An illustration of the proposed language pathway. 74
4.5 Visualized predictions and feature maps of LAVT and ablation models. 82
4.6 Ground truths and predictions of LAVT on examples from the RefCOCO dataset. 83
4.7 Precision-recall curves of LAVT and ablation models on the validation set of RefCOCO. 87
4.8 More predictions and feature maps of LAVT and ablation models. 88
4.9 Success cases and failure cases of LAVT. 89

5.1 An illustration of the task of video object segmentation from referring expressions. 93
5.2 Comparison of different multi-modal feature fusion strategies for video object segmentation from referring expressions. 95
5.3 An overview of the HINet architecture. 97
5.4 Qualitative analysis of the hierarchical feature interaction strategy. 104
5.5 An illustration of the effects of the hierarchical feature interaction strategy and the motion gating branch by examples. 113
5.6 An illustration of the effects of optical flow by examples. 114
5.7 Predictions of the HINet on static objects. 116
5.8 Predictions of HINet on the DAVIS-16 validation set: Part 1. . . . 117
5.9 Predictions of HINet on the DAVIS-16 validation set: Part 2. . . . 118
5.10 Predictions of HINet on the DAVIS-17 validation set: Part 1. . . . 119
5.11 Predictions of HINet on the DAVIS-17 validation set: Part 2. . . . 120
5.12 Predictions of HINet on the A2D test set: Part 1. 121
5.13 Predictions of HINet on the A2D test set: Part 2. 122
5.14 Predictions of HINet on challenging videos from the J-HMDB dataset: Part 1. 123
5.15 Predictions of HINet on challenging videos from the J-HMDB dataset: Part 2. 124

LIST OF TABLES

3.1 Ablation studies of the AD-Net. 46
3.2 AD-Net's object segmentation results on DAVIS-16. 48
3.3 AD-Net's saliency prediction results on DAVIS-16, FBMS, and ViSal. 50
3.4 AD-Net's mean, recall, and decay of region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$, and temporal stability $\mathcal{T}$ on DAVIS-16. . . . 57

4.1 Overall IoU of LAVT on RefCOCO/+/g benchmark datasets. . . 77
4.2 Ablation studies of LAVT. 78
4.3 Experimental verification of the architectural details of LAVT. . . 79
4.4 Fair comparisons between previous methods and the proposed LAVT. 81
4.5 Empirical evaluation of design alternatives for the language pathway. 87
4.6 Mean IoU of LAVT on RefCOCO/+/g benchmark datasets. . . . 88

5.1 The evaluation results of HINet on DAVIS-16. 101
5.2 The evaluation results of HINet on DAVIS-16. 102
5.3 The evaluation results of HINet on A2D and J-HMDB. 103
5.4 Ablation studies of HINet. 105
5.5 A speed-accuracy trade-off analysis of HINet. 106
5.6 Per-attribute results of HINet on the DAVIS-16 validation set. . . 112

1

INTRODUCTION

Computer vision is an important field of artificial intelligence, inasmuch as the ability to derive meaning from vision is an integral part of human intelligence (concerning our species and not individuals). Our day starts as we grab our smartphone and press "stop" on the touch screen to mute the alarm clock, and so has one day of decisions and actions based on visual perception. We freshen up, we get dressed, we take things with us, and we navigate through familiar roads to work or school. In every moment of that portrayed morning routine, our eyes connect us with our environment and let us know where is what, so that we can do what we do.

Computer vision is just the study that aspires to realize this visual perceptual function of ours in computers. While humans directly interpret natural light reflected off objects in the real world, computers work with the digital encoding of that light: 3-tuples organized in rectangular arrays, where the elements of each tuple indicate redness, greenness, and blueness on a scale from 0 to 255, respectively. We call each such 3-tuple a pixel, each rectangular array of pixels an image, and a sequence of images taken continuously in time (think 24 times a second as in a motion picture) a video.

But what does it mean *more specifically*, for a computer to see as a human does? It can mean perceiving depth from stereopsis or motion, it can mean recognizing a familiar face from an image, and it can certainly also mean describing to humans what has been seen from an image or video (such as objects or activities). The scientific community's de facto approach to answering this question can be seen in the dozens of blossoming subdivisions that have

formed, each of which has the goal of realizing certain cognitive function(s) of humans involving vision. Instead of figuring out the actual mechanism of the human brain (hence that of visual perception), which would be too difficult and frustrating, we can summarize and focus on individual tasks that human vision is known to do, and abstract functions from these tasks that we can then aim the computer at learning, usually one at a time but sometimes multiple at once. Not only does this approach provide the periodical feedback that we need to feed our intellectual curiosity, but it delivers results, in time, for solving real-world problems. For instance, depth estimation [1, 2] enables the realistic placement of virtual objects in a physical world [3], facial recognition [4] opens the door to fast and simple access control [5], and automatic image or video captioning [6, 7] makes visual media more accessible to visually impaired people [8].

In this thesis, we focus on one such important task concerning the extraction of the representation of real-world objects from an image or a video, named visual object segmentation. Before introducing this task (and our contributions), it is helpful to first establish some background on the more general concept of visual segmentation.

1.1 VISUAL SEGMENTATION

The word "segmentation" means *dividing a whole into parts*. Visual segmentation refers to the partitioning of a visual representation (*e.g.*, a standalone image or an image from a sequence) into segments, each of which consists of pixels that are *coherent* in either appearance or meaning. This concept can be better illustrated with a few examples from a real-life scenario (Figure 1.1).

At the time of writing (still amid the COVID-19 pandemic), walking to a nearby test site and getting tested for coronavirus is a part of daily life in Shanghai. At three particular moments on a rainy day, while the author of this thesis was in the course of completing this regular task, realizations occurred to him that he was just doing visual segmentation (of different kinds) and



(a) (b) (c)

Figure 1.1: How we want (or want a computer) to segment an image depends entirely on the intended use for the segmentation, *i.e.*, what information we want to extract from the image to our benefit. (a) Segments produced to help the walker stay on the right track (literally and figuratively). (b) Segments produced to keep the walker's feet dry on a rainy day. (c) Segments produced to help the walker take his COVID-19 test. Areas not covered by any colored masks are "don't-care" regions but are still valid segments. Explanations for these examples are in the text below.

critically relying on that fact to fulfill his intentions. When he walked towards the testing site (Figure 1.1 (a)), he had separated what was seen in front of him into areas of regular pavement (blue), tactile paving (yellow), roadside vegetation and structures (cyan), bike lane (purple), and destination (red). Overall, this segmentation oriented him and allowed him to walk safely and efficiently to the destination. To not step into puddles on a rainy day (Figure 1.1 (b)), at times he had to work out the portions of the pavement that were not underwater (cyan). Finally, as he got to the site and wanted to register himself for the test (Figure 1.1 (c)), he had to spot a scanning device (cyan) against which he had known he should display his QR code.

Having looked at these examples, we know that there is a great deal of variability in the way we may segment what we see. If necessary, we can further segment the roadside vegetation and structures (cyan) in Figure 1.1 (a) into

grass, wall, building surface, unidentifiable structure, and tree areas, and we can segment Figure 1.1 (b) into bricks if it has relevance to the application. Then in the academic division, the study of the segmentation of RGB-based visual representations has been put into context and eventually landed upon *tasks* formulated according to the intended application of the segmentation, which include but are not limited to: over-segmentation [9], contour detection [10], salient region detection [11], motion segmentation [12], semantic segmentation [13], instance segmentation [14], panoptic segmentation [15], video object segmentation [16], object co-segmentation [17], *etc*. And it is often the case that several tasks are related to each other in certain aspects, such as having overlap in their objectives or yielding results that are complementary.

1.2 VISUAL OBJECT SEGMENTATION

In this thesis, the type of segmentation that we are interested in involves discovering representations from images or videos that correspond to physical entities in the real world, which most frequently are things described by countable nouns (*e.g*., a toothbrush, a person, and a car) and occasionally are things identified by uncountable nouns (*e.g*., water, grass, and sky). We refer to all these entities as "objects" with a small generalization from the word's most natural meaning.

At present, the typical setup under which to study visual object segmentation is established on static images with a condition, that a categorization of plausible objects is made available. Such is the setup adopted in the tasks of semantic segmentation [13], instance segmentation [14], and panoptic segmentation [15]. Given a superset of categories which have "meanings" (such as bicycle, human, sky, furniture, *etc*., plus a generic "don't-care" category), semantic segmentation partitions the image into regions corresponding to the categories, instance segmentation partitions the image into instance-aware regions that correspond only to categories permitting the differentiation of instances, and panoptic



Figure 1.2: Three popular formulations for the problem of visual object segmentation, which are based on a categorization of real-world entities. They extract information from static images and rely on the pre-definition of a set of semantic categories (*e.g.*, 20, 80, and over 1000 categories for the PASCAL VOC [23], COCO [24], and LVIS [25] datasets, respectively). There are two generic limitations with this setup: Algorithms cannot deal with entities that fall outside the range of the pre-defined categories, and temporal information, especially that of moving entities, cannot be captured by static images and is thus not leveraged for segmentation. Image source: [26].

segmentation unifies the two tasks (Figure 1.2). In what is described above, the notion of "instance" refers to an individual of a category, and instance segmentation is different from semantic segmentation mainly in that it requires the separation of individuals within each class.

This setup has been undoubtedly successful, as publications grew rapidly over the past years and a great many influential models [13, 14, 18] were born out of research on those problems. Still, our essential motivation is to extract representations of physical entities in the real world from visual data, and from this perspective, it helps us to regard more realistic situations. The objects we encounter in real life are *arbitrary*, meaning that they are not confined to the fixed number of categories that we may define. Therefore, the algorithm should just be able to segment out *any* object as opposed to only objects of certain categories. Moreover, our experience in this world is continuous (in time), which means that we perceive objects as evolution of shapes, positions, and appearances (mainly due to the movement of the objects or ourselves) instead of snapshots at certain instances of time. From a practical point of view, the continuous views of an object provide us with information on motions and appearance correspondences, which can help us discern the object from other things [17, 19–22].

To address the first limitation, in this thesis we explore language-guided visual object segmentation, *i.e.*, rather than conditioning object segmentation

on a finite set of pre-defined semantic categories, we aim to segment out any object in accordance with a natural language description. To address the second limitation, we investigate object segmentation in the context of videos, where the modeling of temporal cues is essential for accurately segmenting out objects in each frame of the video.

Our endeavors in the above two directions have led to our contribution of methods to three important tasks in computer vision, which we introduce as follows (and detail in the following subsections respectively): automatic video object segmentation, image segmentation from referring expressions, and video object segmentation from referring expressions. The first, the second, and the third tasks present challenges in exploiting temporal, linguistic, and both types of cues for visual object segmentation, respectively. On a high level, these three tasks are dense prediction problems and bear similarities with semantic segmentation—they require classifying each pixel as belonging to the "target" category or the "background" category. Therefore, our solutions to them share some common ingredients, including incorporating dense feature-encoding vision networks and convolutional classifiers in the architectural design and adopting a fully supervised, end-to-end learning strategy. At the same time, each task presents unique challenges, and we make innovations specific to the context.

1.2.1 *Automatic video object segmentation*

The task of automatic video object segmentation [16, 27–29] is also referred to as unsupervised video object segmentation or zero-shot video object segmentation (Figure 1.3), where the adjectives "automatic," "unsupervised," and "zero-shot" denote that there is no human intervention at inference time for guiding segmentation, as instead is the case in semi-supervised (or, one-shot) [30, 31] and interactive [32] video object segmentation. It aims at predicting a pixel-wise mask, in each frame of the video, which precisely covers an object or a group of objects that are salient. In practical terms, the current definition of an



Figure 1.3: Automatic video object segmentation concerns the prediction of pixel-wise masks for salient, moving objects in a video.

object being "salient," according to the existing benchmarks [16, 27–29], is being sufficiently large and moving independently from the background, hence likely to attract a viewer's attention. This task yields value to a variety of real-world applications such as autonomous driving [33–35], robotic manipulation [36], video surveillance [37], and video editing [38].

As a video records the dynamic changes of a scene, the accurate segmentation of objects can be made challenging by factors such as occlusion, scale variation, appearance change, motion blur, *etc*. To overcome these difficulties, most methods in the past have relied on recurrent neural networks (RNNs) [39, 40] or optical flow [40–47] to model pixel correspondences and motion cues, based on which to disseminate features or predictions across adjacent frames.

While this approach has the potential of adapting to position and appearance changes of the objects, it has several drawbacks due to the limitations of RNNs or optical flow. RNNs rely on the training technique of truncated backpropagation through time, which limits their long-term modeling capability [48]. And in spite of the development of gating mechanisms in popular structures such as LSTMs [49] and GRUs [50], vanishing or exploding gradients still cause problems for the training of RNNs [51, 52], which can be alleviated but not eliminated by techniques such as gradient clipping and re-scaling [53]. Optical flow, on the



Figure 1.4: Given an image and a description of objects of interest, the objective is to segment out the region corresponding to those objects. This is the task of image segmentation from referring expressions, or referring image segmentation.

other hand, generally contains inaccuracies which can cause errors to accumulate when it is used for guiding feature or prediction propagation across frames. The deterioration of performance over time of methods based on optical flow or RNNs has been widely discussed [44, 54–56]. In particular, in [56] (and our study), it is shown that static foreground-background segmentation models can perform on a par with some of the state-of-the-art methods. These observations suggest that alternative temporal modeling strategies should be explored.

1.2.2 *Image segmentation from referring expressions*

Instead of relying on pre-defined categories to segment out objects from an image [13–15], sometimes it is more useful to segment out objects according to human natural language [57, 58], as this allows the segmentation of *any* object and the dynamic specification of targets based on the need of the intelligent system (Figure 1.4). As this task involves the joint reasoning over image and text content, it is considered a *multi-modal* task, and the key challenge is in aligning linguistic meanings with corresponding visual clues.

In the past, methods addressing this task mainly aim at developing powerful *multi-modal feature fusion modules* [58–68] that can capture dependencies across the two domains, which are applied on top of two independent feature encoders dedicated to embedding the image and the text inputs, respectively. This approach works, as the segmentation accuracy has improved consistently over

the past years. Very recently, the Transformer model [69, 70] has established its status as the most powerful mechanism for addressing multi-modal feature fusion, as several methods employing Transformers to fuse visual and linguistic features attain the highest accuracy on this task [60, 71]. Having originally been designed for the machine translation and language modeling tasks [69, 70], the Transformer model consists of stacked *attention* layers which compute the features at each position of the output as an aggregated sum of features at all positions of the input, based on pair-wise similarities; this approach of feature learning based on global correlation modeling has been shown generalizable to the vision domain and is starting to dominate both the natural language processing field [70, 72, 73] and the computer vision field [74–78].

The question we need to ask is that if the Transformer has been shown effective at aligning cross-modal features at a late stage, *i.e.*, after the independent encoding of the image and the text inputs, then could we possibly benefit from an earlier stage of cross-modal feature fusion via the Transformer? One potential option is to leverage a vision Transformer backbone network for jointly encoding the image *and* aligning the intermediate representations with linguistic features, under the rationale that more Transformer layers (in the order of ten times as many) can participate in excavating helpful multi-modal context, thus leading to more accurate alignment. If successful, this approach offers "free" benefits in the sense that the Transformer layers in the vision backbone network already exist (for the purpose of image encoding) and we simply exploit them as the workhorse also for conducting cross-modal feature fusion, without having to rely on dedicating additional, heavy structures to this purpose.

1.2.3 *Video object segmentation from referring expressions*

In the task of video object segmentation from referring expressions (Figure 1.5), the problems of cross-modal alignment and temporal information modeling must both be addressed. Although this task is connected with previous ones that

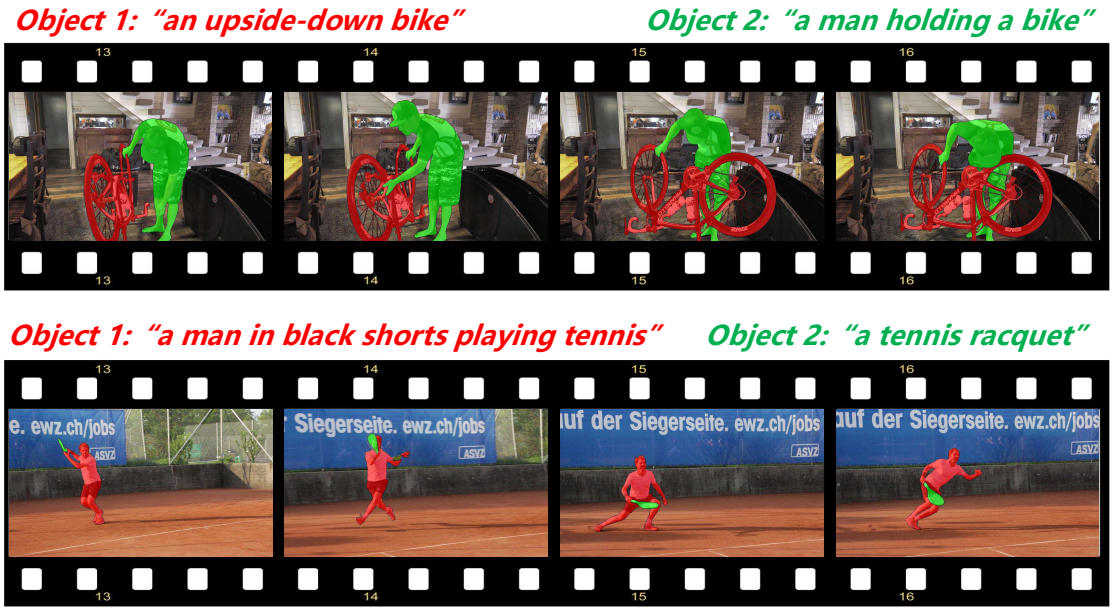


Figure 1.5: Given a video and a referring expression, the task of video object segmentation from referring expressions is to segment out the object referred to by the expression in each frame of the video. It can be viewed as the task of video object segmentation with language initialization, or an extension of the referring image segmentation task to the temporal domain.

we have introduced, the development of methods for addressing it has followed a different path due to its origin in the problem of pixel-level understanding of the actors and their actions in videos [79]. As a result, the state-of-the-art methods have some unique characteristics and face their own set of challenges.

While various multi-modal feature fusion schemes [79–84] have been developed for tackling this problem, by comparison, they are generally less sophisticated in design. One common issue among the fusion mechanisms is the neglected use of the multiple levels of visual representations inherently existing in modern-day convolutional neural networks (CNNs), in spite of their wide-spread adoption in image segmentation models (including referring image segmentation models). This issue is acute if we look at the adjacent fields of instance and semantic segmentation, where the exploitation of multi-level features for dealing with scale variations has been common, and if we examine this problem's native need for addressing the cross-modal alignment of concepts that have different levels of complexity (for instance, colors and categories).

Another general issue faced by existing methods lies in the exploitation of motion cues which, by comparison with its development for automatic video object segmentation, has also been quite neglected. For the most part, temporal dependency modeling is delegated to a 3D CNN encoder [85, 86], which is effective at capturing action information but less effective at retaining accurate spatial details [87, 88].

1.3 CONTRIBUTIONS AND THESIS OUTLINE

In this section, we outline the contributions of this thesis, which take place in the novel and effective methods proposed for addressing the problems of automatic video object segmentation, image segmentation from referring expressions, and video object segmentation from referring expressions. Collectively, these problems constitute a study of visual object segmentation in scenarios where temporal and linguistic cues come in useful, and the proposed methods make innovations in the architectural design of deep neural networks to effectively exploit such cues for object segmentation.

Each of Chapters 3 to 5 originates from a peer-reviewed paper that was published in the proceedings of an academic conference. We edit the original conclusion section from each paper to provide more discussion on the intuitions behind the proposed method as well as the method's limitations.

An overview of the content of the individual chapters is the following.

- Chapter 2 gives an overview of literature on the topics of video object segmentation (Section 2.1), image segmentation from referring expressions (Section 2.2), and video object segmentation from referring expressions (Section 2.3). We describe the progress brought forward by recent methods as well as challenges faced by them.

- Chapter 3 [89] presents our first contribution—a novel method for tackling the problem of automatic video object segmentation. Unlike previous

methods which rely on optical flow or recurrent neural networks to sequentially propagate features or predictions across frames, our method establishes pixel correspondences between a reference (the "anchor") frame and the current (the "target") one, and directly propagates features between these two frames. Crucially, by allowing pixel correspondences to be learned arbitrarily far in time without depending on intermediate frames, this approach sidesteps the limitations of the sequential modeling approach, *i.e.*, a tendency to accumulate error and a favor for short-term dependencies. As a result, our method is able to achieve high robustness over time and despite its simplicity, obtains accuracy competitive with or superior to the state of the art on three benchmarks at the time of publication.

- Chapter 4 [90] presents our second contribution—a novel method for tackling the problem of image segmentation from referring expressions. In the past, the learning of cross-modal feature alignment, central to this problem, has been largely confined to a stage that comes after the independent encoding of the image and the text inputs. Our proposed method relocates this key process to the phase of image encoding, showing that significantly better cross-modal alignment can be achieved via the early fusion of linguistic and visual features in intermediate layers of a vision Transformer network. The main procedure involves "injecting" linguistic cues into the vision Transformer network during its forward pass. To this end, we develop a dense attention mechanism for collecting pixel-specific linguistic cues, and a self-adaptive gating mechanism for adjusting their values when they are sent to the vision Transformer network. This approach is able to exploit the correlation modeling power of the vision Transformer network for cross-modal feature fusion, and as a result, unlike previous methods, our method can employ a standard, light-weight predictor to produce the final segmentation. Despite its simplicity,

our method obtains accuracy superior to the state of the art on three benchmarks by the date of publication.

- Chapter 5 [91] presents our third contribution—a novel method for tackling the problem of video object segmentation from referring expressions. The core idea of our approach is to leverage the feature pyramid inherently existing in a convolutional neural network (CNN) and integrate linguistic features with visual features from different levels of the CNN. This approach allows the cross-modal alignment of meanings on multiple scales and at different levels of semantic complexity, which leads to more accurate pixel-wise referral compared with previous methods that perform single-level alignment. Moreover, instead of relying on 3D CNNs for modeling temporal cues, we integrate optical flow information with the multi-level, cross-modal feature maps. This avoids the spatial misalignment problem of 3D CNNs, while at the same time enabling us to make online predictions without requiring long video clips as inputs. Despite its simplicity, our method produces results superior to the state of the art on four benchmarks at the time of publication.

- In Chapter 6, we conclude this thesis by summarizing the main findings and suggesting directions for future research.

1.4 PUBLICATIONS

The three central chapters of this thesis originate from the following publications. Asterisks indicate joint first authorship.

- Zhao Yang*, Qiang Wang*, Luca Bertinetto, Song Bai, Weiming Hu, and Philip H.S. Torr. Anchor diffusion for unsupervised video object segmentation. In *ICCV*, 2019.

- Zhao Yang*, Jiaqi Wang*, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip H.S. Torr. LAVT: Language-aware vision Transformer for referring image segmentation. In *CVPR*, 2022.

- Zhao Yang*, Yansong Tang*, Luca Bertinetto, Hengshuang Zhao, and Philip H.S. Torr. Hierarchical interaction network for video object segmentation from referring expressions. In *BMVC*, 2021.

Other publications were made during the course of this PhD.

- Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip H.S. Torr. Semantics-aware dynamic localization and refinement for referring image segmentation. In *AAAI*, 2023.

- Zhao Yang*, Qiang Wang*, Song Bai, Weiming Hu, and Philip H.S. Torr. Video segmentation by detection for the 2019 Unsupervised DAVIS Challenge. In *CVPR Workshops*, 2019.

- Qiang Wang, Yi He, Xiaoyun Yang, Zhao Yang, and Philip H.S. Torr. An empirical study of detection-based video instance segmentation. In *ICCV Workshops*, 2019.

- Zhao Yang, Song Bai, Li Zhang, Philip H.S. Torr. Learn to interpret Atari agents. *Technical report*, 2018.

2

LITERATURE REVIEW

2.1 VIDEO OBJECT SEGMENTATION

The problem of video object segmentation (VOS) is tackled by the computer vision community mainly in semi-supervised and unsupervised settings, which are defined by the level of supervision provided at *test time*. This definition was popularized with the publication of DAVIS [16], a benchmark and yearly challenge for VOS. When we introduced VOS in Chapter 1, we replaced the word "unsupervised" with "automatic," to avoid any chance of misinterpretation that ground-truth labels are not available at *training time*, which would be what "unsupervised" means outside the context of VOS. We make this clarification here and continue with the naming tradition of VOS.

Semi-supervised VOS (also called one-shot VOS) methods are provided with a pixel-wise mask identifying the target object in the first frame of a video, and based on the current literature, there are generally two ways to utilize this mask.

The first is to supervise the training of a model *online* (*i.e.* at test time) with this mask, so that the model can learn the appearance of the target object, and specialize in separating *this* object from everything else (*i.e.*, the background). This approach is intuitive and underlies many state-of-the-art methods [56, 92–97]. It requires, first, the offline training (*i.e.*, pre-training) of a base model that can perform generic binary segmentation (*i.e.*, having some notion of "objectness"), and then, the online fine-tuning of the base model using the first frame and the mask, by exploiting data augmentation techniques such as affine transformation, thin plate spline, color jittering, Gaussian blur, copy-pasting

of the object (onto different backgrounds), *etc*. The aim of the above artificial transformations is two-fold: first, to avoid model overfitting by introducing variance to the training data (which would otherwise only be a single datum), and second, to encourage the model to learn helpful invariances, since some of the transformations are similar to real online changes such as translations, non-rigid deformations, motion blurs, generic background changes, *etc*. However, artificial transformations do not fully capture natural transformations in a video. To address this issue, Maninis *et al*. [96] and Voigtlaender *et al*. [56] propose to *self-supervise* the fine-tuning process, by employing predicted masks as pseudo-ground truths to continuously update the model as it makes online predictions.

The second way to utilize this mask, also quite intuitively, is treating it as a reference mask and propagating it to the rest of the frames [54, 55, 93–95, 97–100]. This requires the learning of a function that can establish dense associations between frames, based on which to propagate the mask. Many different methods have been developed based on this idea. For instance, Perazzi *et al*. [97], Khoreva *et al*. [101], and Oh *et al*. [55] propose to incorporate the initial ground-truth/last predicted mask as an additional channel of the input at each forward pass, Khoreva *et al*. [101] and Luiten *et al*. [95] leverage optical flow for directing feature propagation across frames, Li *et al*. [94] and Luiten *et al*. [95] exploit instance embeddings pre-trained for object re-identification, Chen *et al*. [54] learn a metric space for pixel embeddings, which is then used to establish associations between pixels across frames, Cheng *et al*. [98] suggest to individually track object parts from the first frame with a visual object tracker [102] and then aggregate them according to their similarity with the initialization mask, and Bao *et al*. [99] define and optimize a spatial-temporal energy function on a random field composed of all pixels in the video.

When aiming at very high segmentation accuracy, methods generally adopt both of the above strategies, performing online fine-tuning while also implementing some mechanisms of mask propagation [56, 93–97]. As online fine-tuning

can take up to several minutes per video, a lot of methods renounce it and adopt only mask propagation to obtain a higher inference speed [54, 55, 98–100, 103].

Unsupervised VOS methods, instead, cannot rely on any supervision at test time. As there is no initialization at test time, to reduce ambiguity in the problem setup, unsupervised VOS generally makes a simplifying assumption, that there is a single object (or a group of spatially adjoining objects) that is salient (*i.e.*, attracting the attention of the viewer). In practical terms, according to the benchmarks (*e.g.*, [16, 27–29]), a salient object is usually one that is large, centered, and independently moving.

Since saliency is strongly correlated with motion, unsupervised VOS methods are often based on optical flow. The purely optical flow-based MP-Net [47] discards appearance modeling and casts segmentation as foreground motion prediction, an approach which deals poorly with static foreground objects. To address this problem, several methods (*e.g.*, LVO [40], FusionSeg [42], SegFlow [41], MotAdapt [46], and MBN [45]) suggest to integrate appearance-based and optical flow-based features together, leading to variations of the "two-stream model" presenting two dedicated parallel branches. Some other methods which do not assume a two-stream architecture leverage optical flow fields to warp features from the previous frame to the next one (*e.g.*, [43]), or compute motion saliency scores for tracking regions (*e.g.*, [43, 44]). The drawbacks of these methods are threefold. First, flow estimation networks are typically trained on synthetic datasets and can thus result in poor performance when deployed in the real world. Second, while modeling long-term temporal dependencies is critical for adapting to significant online changes, the vector fields can only model short-term one-step dependencies. Targeting this issue, Tokmakow *et al*. [40] propose to extend the horizon spanned by optical flow-based features by employing a convolutional gated recurrent unit [50]. Third, vector fields cannot distinguish foreground and background objects when they move in a synchronised fashion (*e.g.*, the cars in a traffic jam). Li *et al*. [45] attempt to address this issue by employing a bilateral network for detecting

the motion of background objects. Our investigations with a much simpler appearance-based approach show that optical flow may not be an essential component of unsupervised VOS systems.

RNN-based models are often challenged by the problems of exploding and vanishing gradients [51, 52], which limit their long-term modeling capability. Among the methods that make use of recurrent connections, Song *et al*. [39] propose a novel convolutional long short-term memory [49] architecture, in which two atrous convolution [104] layers are stacked along the forward axis and features are propagated in opposite directions, while Tokmakow *et al*. [40] implement a recurrent visual memory system to capture the evolution of objects over time.

Recently, it has been shown [44, 55, 56] that both recurrent and optical flow-based methods suffer a significant deterioration in the quality of their predictions over time. This has motivated several approaches that tackle video object segmentation by simply learning similarities between pixel embeddings (*e.g*., [44, 45, 54, 105]). These methods first select a set of seed pixels that are most likely to belong to the foreground object and then classify all other pixels based on their similarities to these seeds, for instance by thresholding or by propagating labels between neighbors. Fathi *et al*. [105] adopt this approach for semantic instance segmentation, in which the pairwise pixel similarity function measures the likelihood of two pixels belonging to the same instance. IET [44] extends this concept to video sequences. Similarly, it selects a set of foreground and background seeds for each frame and organizes them into tracks. It then segments each frame individually based on pixel similarities with the foreground and background seeds. Note that IET utilizes pre-trained instance embeddings. MBN [45] extends IET with a bilateral filtering network that filters false-positive foreground predictions using optical flow features and an energy minimization procedure on a graph of seeds sampled from a few consecutive frames. When segmenting frame t, MBN classifies each pixel by

assigning it the label of the seed (sampled from frames $t-1$, t, and $t+1$) with which it has the smallest embedding distance.

The main drawback of these methods is in the complexity involved in the procedures of seed selection, ranking, and classification, critical for achieving good performance. Moreover, these algorithms also depend on multiple scores such as motion saliency and objectness that need to be carefully calibrated and combined into one final metric.

2.2 REFERRING IMAGE SEGMENTATION

Referring image segmentation is the task of image segmentation from referring expressions. A referring expression is a piece of text that describes, in natural language (*i.e.*, human language), things of interest in an image. And the objective is to segment out the things that are referred to.

The idea of language-guided image segmentation can be traced back to the work of Cheng *et al*. [57], where an interactive image segmentation system was proposed, to allow for the parsing of an image based on user-supplied verbal commands. While being the first of its kind, the proposed system has limited capacity, as the verbal commands follow canonical syntactic forms and work with a closed set of vocabularies, including 16 object class words and 27 attribute words divided into sub-groups (such as material, surface, color, and position).

The current formulation of the problem of referring image segmentation based on free-form expressions was proposed by Hu *et al*. [58] in 2016, at a time when the development of large-scale datasets (*e.g.*, [106–109]), instance/semantic segmentation systems (*e.g.*, [13, 14]), and language modeling techniques (*e.g.*, [53, 110, 111]) has reached some point of maturity and was gaining momentum. Referring image segmentation (RIS) has since been a popular research topic in the computer vision community.

Developments in state-of-the-art methods for the problem of RIS are thriving and taking paths in two directions, one learning from how we tackle the problem

of *semantic segmentation* and the other learning from the experiences of tackling *instance segmentation*. We introduce both types of methods in the following.

Hu *et al*. [58] proposed the first of the kind of algorithms based on the fully-convolutional architecture [13]. The fully-convolutional approach casts RIS as a semantic binary segmentation problem, where the "semantics" of interest is defined by the referring expression. The pipeline involves three steps: (1) encoding the image and the text inputs with a vision encoder (which is a convolutional network in early methods and vision Transformer [74, 77, 78] in recent ones) and a language encoder, respectively, (2) embedding the two types of features into a common feature space, leading to a set of multi-modal feature maps that highlight the target objects, and (3) segmenting the multi-modal feature maps via a convolutional classifier.

A vision encoder is a deep neural network that has been pre-trained for solving certain important visual recognition tasks such as image classification (*e.g.*, [112]), object detection (*e.g.*, [113]), and semantic segmentation (*e.g.*, [104]), which is capable of producing features (from the raw input) generalizable to downstream tasks. Such a network is usually employed as an "encoder" or "backbone" component in a larger task-specific model to bootstrap training and improve the model performance. Recently, there is a surge of interest in the unsupervised/self-supervised learning of representations, which attempts to alleviate the dependency on costly manual labels [114–116] for training such networks.

Similarly, a language encoder is a deep neural network that is employed for extracting helpful features from text inputs. Earlier examples of a language encoder include various types of recurrent neural networks (*e.g.*, GRUs and LSTMs), convolutional networks, and multi-layer perceptrons (MLPs). To use these networks, words are first converted to high-dimensional embeddings learned from large-scale corpora [110, 111, 117], which are typically kept fixed during training, while the language encoder is trained from scratch. More recent and more powerful language encoders are typically based on the

Transformer architecture [69] and are pre-trained for solving various language modeling tasks [70, 118, 119]. During pre-training, the word embeddings and the Transformer layers are jointly optimized, while in downstream tasks, typically only the Transformer layers are optimized. The concept of a language encoder in the context of this chapter is closely related to the literature on representation learning for natural language processing, and interested readers can refer to [120] for an introduction to this topic.

With two types of features sharing information represented in different modalities, and the objective being pixel-wise referral, how to highlight positions in the image based on feature correspondences becomes a question of central importance. Correspondingly, much effort has been devoted to research on the multi-modal feature fusion mechanism. To this end, Hu *et al*. [58] suggest concatenating sentence features with image features from the top layer of the vision encoder and applying an MLP projection to the concatenated features. While being effective, this simple baseline approach does not take into account the "shape" of the data, *e.g*., an expression is a sequence of interdependent words and an image can be represented by a feature pyramid. This has motivated the several methods that take a recurrent approach to feature fusion (*e.g*., [61, 62, 65, 121]). Taking inspiration from how humans tackle visual grounding tasks, Liu *et al*. [61] propose to integrate linguistic and visual features step by step in a word-reading order, while Li *et al*. [121] suggest recurrently integrating linguistic features with visual features from different levels of the image feature pyramid to capture multi-scale semantics. Although the recurrent strategy is successful, at each time step the linguistic and visual features are still combined via concatenation (before the concatenated features are sent to a multi-modal RNN), and this proves to be a less effective way of measuring the "similarity" between words and image regions, as we see in later work that adopts *attention*-based fusion strategies.

The modern concept of attention was proposed in an attempt to address two prominent issues faced by the then state-of-the-art methods addressing the

task of neural machine translation, which is the task of translating a sentence from one language to another language via neural networks. Back then, the most promising approach was the "seq2seq" model [50, 53], based on an RNN encoder-decoder architecture, where an RNN encoder first recurrently encodes the source sentence into a fixed-size vector representation, and then an RNN decoder recurrently decodes from this vector the translated sentence word by word. This method faced two fundamental challenges: One is the fixed capacity of the RNN hidden state (a feature vector) for representing arbitrarily long input sentences, and the other is the long propagation path of the gradients from the output words to the input words.

To understand the first challenge, consider this passionate quote from Raymond Mooney, "[y]ou can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector" (and fill in the masked words with your language model of informal English, he said) [122], which is to suggest that a fixed-size representation may simply be not enough to hold all the information in the sentence. And to understand the second challenge, consider the path of the gradients backpropagated from an output word to an input word, which is purely sequential and chained through time. This makes the learning of dependencies between output words and input words inherently difficult (for instance, due to vanishing or exploding gradients). In addition, there lacks a mechanism that exploits the natural correspondences between output words and input words, for instance, the fact that the prediction of one word in the target language typically relies on some words in the source language more than it does other words, due to varying degrees of correspondence in their meanings (*e.g.*, "je" depends on "I" but not "you" in the translation of "I love you" in English to "je t'aime" in French).

Observing these issues, attention was proposed as a memory module equipped with a correlation-based "querying" mechanism, which is a new way to condition representation decoding on the encoded features [123]. It maps the representation of each output word to a combined representation of all the input words by

learning a distribution based on one-to-many, pair-wise dot-product similarities (the original formulation in [123] actually used an MLP instead of dot-product for modeling the correspondence). This greatly shortens the backpropagation path of the gradients and also solves the fixed-size representation issue (as the length of the memory grows with the number of input words). But more importantly, this method of correspondence (or equivalently, correlation, association, similarity, relevance, *etc.*) learning proves to be extremely important and effective for addressing natural language translation, a sequence transduction problem.

Later in their seminal work [69], Vaswani *et al*. made two critical improvements based on this initial formulation and usage of the attention mechanism. The first is the formulation of the concepts of *key*, *value*, and *query* in the attention mechanism, and the second is the expansion of the use of the attention to the encoder and the decoder (in contrast to only using it for mapping representations between the decoder and the encoder), thus removing the RNNs and leading to an all-attention architecture (named the Transformer) for addressing the task of neural machine translation.

Later on, the attention mechanism produced even greater impacts with the publication of BERT [70], in which it is shown that strongly generalizable representations can be obtained from the Transformer encoder by training it on the masked language modeling and next sentence prediction objectives. Henceforth, attention has found success in a broad range of tasks that would benefit from correlation modeling (*e.g.*, [74, 78, 124–126]), and in some of these cases, the importance of correlation modeling was not fully realized before attention has demonstrated it for the relevant tasks.

Since the central challenge in RIS is precisely one that involves correlation modeling, between vision and language, the RIS community has evolved a great interest in the potential of *cross-modal attention*-based fusion strategies (*e.g.*, [59, 63–68, 127, 128]). Similar to how attention works in the context of machine translation, cross-modal attention essentially aims at conditioning the classification of each image region (centering on a pixel) on words that

relate to the region, based on one-to-many, pair-wise dot-product similarities. There can be many ways to realize this, and various methods have developed many different forms of cross-modal attention. For instance, Ye *et al*. [64] apply self-attention on concatenated word and pixel features, Hu *et al*. [66] and Feng *et al*. [128] develop different kinds of bi-directional cross-attention modules to align features both in the image-to-text direction and the text-to-image direction, Huang *et al*. [67] establish relations between image regions and different groups of attribute words via multi-modal graph reasoning, Hui *et al*. [68] exploit syntactic structures of language for guiding multi-modal context aggregation, Luo *et al*. [59] devise a grouped attention layer and put multiple such layers in a cascade for iteratively refining cross-modal mappings, and Ding *et al*. [60] abstract multiple interpretations from the sentence, and use them as queries in a Transformer encoder-decoder [69] for aggregating interpretation-specific visual context.

The above constitutes our summary of work that approaches RIS from a semantic segmentation point of view, and in what follows, we introduce methodologies that take inspiration from instance segmentation systems [71, 129–131]. The first is the two-stage approach, which consists of first proposing candidate regions and then matching the referring expression to these regions [129, 130]. The general philosophy here is to treat RIS as a search problem based on an evaluation criterion: After having evaluated the compatibility of the given sentence with all regions that are likely to contain an object, we can select the best match as our prediction. Region proposal is typically tackled straightforwardly via an off-the-shelf instance segmentation network (*e.g*., Mask R-CNN [14]) and the focus of such two-stage methods lies in the cross-modal evaluation phase.

Yu *et al*. [129] propose a matching protocol that evaluates a region based on the weighted average of three types of matching scores: subject, location, and relationship, where the individual matching scores are computed via dedicated modules that evaluate the match between a region and a subject phrase, a location phrase, and a relationship phrase, respectively. Here, the

phrases are parsed from the input sentence by a language attention network, which also predicts the averaging weights of the three types of scores. In a similar vein, Liu *et al*. [130] also evaluate a region based on its matching scores with individual phrases parsed from the sentence. But more effectively, they decompose the sentence into a tree structure by leveraging a dependency parsing tool, and aggregate scores by traversing the tree from the bottom level to the top level. The finer-grained partition of the sentence and the syntactic dependencies provide a more adequate way of representing the sentence and allow for more accurate cross-modal matching and region selection, leading to better segmentation results. Overall, a fundamental limitation of the two-stage approach is that region proposal is detached from cross-modal matching, and as a result, proposals cannot be learned from language, which leads to a low recall rate [132].

The second kind of methods tap into the recent development of one-stage object detectors based on the Transformer architecture (*e.g.*, DETR [75]). Kamath *et al*. [131] and Li *et al*. [71] each propose a model based on the Transformer encoder-decoder architecture [70] that can address RIS in an end-to-end fashion. Kamath *et al*. [131] design their model for text-modulated object detection in images (*i.e.*, predicting a bounding box for every object that is mentioned in the text input), and aim for it to be a general system that can be fine-tuned (and modified to have the appropriate prediction heads) to solve a variety of downstream tasks, including phrase grounding, referring expression comprehension (REC), RIS, and long-tailed object detection in a few shot setting. For this reason, text features and image features are jointly embedded in the Transformer encoder while the learnable query embeddings in the decoder are designed the same way as in DETR [75]. When fine-tuned for the task of RIS, the model employs a segmentation head. Conversely, Li *et al*. [71] aim for their model to be able to jointly address the tasks of REC and RIS. Therefore, their model employs text features as query embeddings in the Transformer decoder, and has two heads in parallel for box prediction and segmentation, respectively.

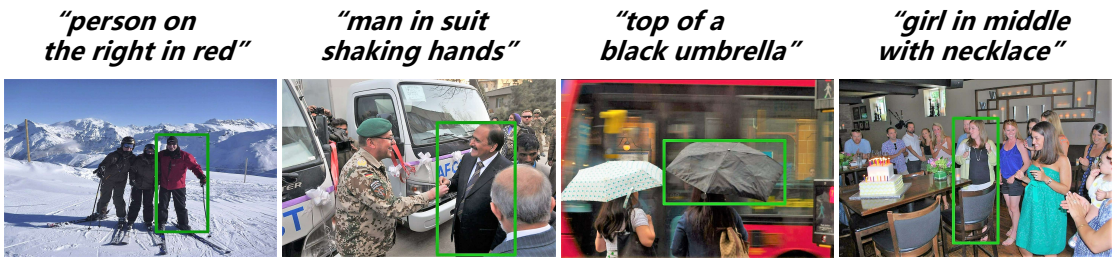


Figure 2.1: The task of referring expression comprehension asks to put a bounding box on the region referred to by the given expression. The naming rationale is that the bounding box is one way of representing an algorithm's "comprehension" of the referring expression. A task opposite to this is referring expression generation (see [109]).

Overall, the two-stage approach and the one-stage approach based on Transformers that address RIS are very closely related to the literature on REC (Figure 2.1), a task similar to RIS that requires a bounding box as the prediction. For this reason, many one-stage and two-stage models designed for addressing REC have the potential to be adapted with a small amount of effort to address RIS. Among such methods, the one-stage models typically exploit dense anchoring mechanisms (such as ones in [133, 134]) or the Transformer architecture, and the two-stage models sometimes rely on unsupervised proposal methods such as [135, 136]. To keep this review relevant to RIS, we omit the introduction to these models and recommend [132, 137, 138] to readers for more information.

2.3 Video Object Segmentation from Referring Expressions

The problem of video object segmentation from referring expressions (VOSRE) was first proposed by Gavrilyuk *et al.* [79] as an extension of the problem of actor-action segmentation in videos [139] from the *fixed vocabulary* domain to the *natural language expression* domain. Actor-action segmentation is the task of segmenting out, in each frame of the video, an *actor* (*e.g.*, adult, baby, ball, bird, *etc.*) that is performing an *action* (*e.g.*, climb, crawl, eat, fly, *etc.*), where the actors and the actions, in Xu *et al.* [139]'s original formulation, are defined by 43 plausible combinations of 7 actor classes and 9 action classes, and in

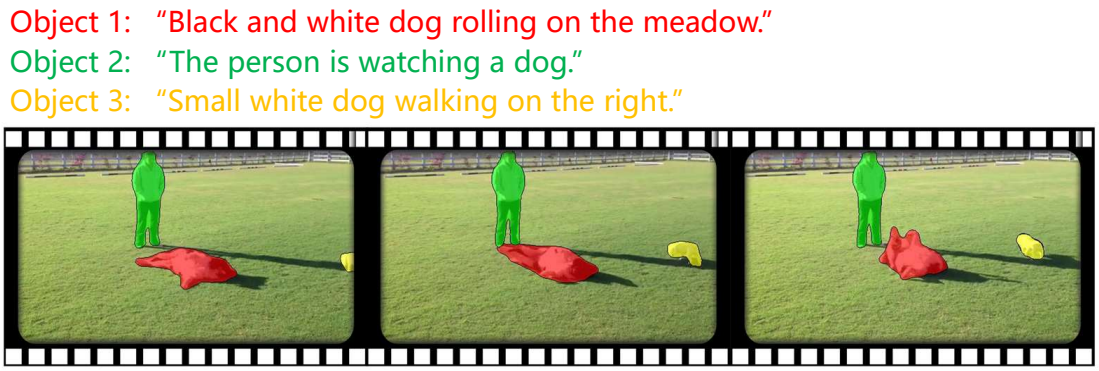


Figure 2.2: The task of actor-action segmentation from a sentence is similar to video object segmentation from referring expressions, but the input expression must describe the action that an actor is performing.

Gavrilyuk *et al.*'s extension, are freely specified by natural language descriptions (see Figure 2.2). Inspired by the work of Gavrilyuk *et al.*, Khoreva *et al.* [140] and Seo *et al.* [84] extend the problem of video object segmentation (introduced in Section 2.1) to the natural language domain, by annotating objects in the DAVIS-16 [16] and DAVIS-17 [30] datasets, and objects in the YouTube-VOS dataset [31] with natural language descriptions, respectively.

In the following, we introduce the existing methods that tackle the problem of VOSRE, where we organize most methods into three groups based on the type(s) of vision encoders (described in Section 2.2) they are designed to work with: image encoders, video encoders, and both types of encoders.

Methods adopting an image encoder (*e.g.*, ResNets [141]) intend to capture temporal dependencies primarily via association functions that operate on multiple frames, similar to what we have described in Section 2.1 for semi-supervised VOS methods. Seo *et al.* [84] leverage a space-time attention network [142], which adopts the target frame as the query and historical frames as the key and the value for capturing temporal changes of the target object. The cross-modal referral of the target object is achieved via a cross-modal self-attention network [143] that we introduced in Section 2.2. The temporal features and cross-modal features are combined via addition in a segmentation decoder. While this approach models both linguistic and temporal dependencies, it does

so via two independent fusion networks, which leads to a minimum of two rounds of inference for each video at test time. The first round involves only the cross-modal fusion network and predicts independently one mask for each frame. After a most-trusted mask (*i.e.*, one having the highest average pixel prediction scores) is selected, the second round involves the full network and propagates this mask to the rest of the frames. The process in the second round can be repeated many times to iteratively refine the results. Khoreva *et al*. [140] adopt a similar approach. Their method first leverages an off-the-shelf referring expression comprehension model to predict, for each frame independently, a set of candidate bounding boxes accompanied by matching scores. Then for each frame, the candidate bounding boxes are re-ranked based on a measure of temporal consistency, which by definition is proportional to a box's total overlapping area with all other boxes in the video, and inversely proportional to the box's total temporal distance with all other boxes in the video. Finally, a mask is predicted for each frame, from a concatenated input comprising the original frame, its optical flow magnitude, and the binary mask of the highest-ranking box for that frame, by a network trained for saliency segmentation.

Methods adopting a video encoder (*e.g.*, C3D [85], I3D [86], and P3D [144]) mainly aim at capturing temporal dependencies via 3D convolutions, an approach well motivated for the problem of action recognition [86] as it requires the modeling of spatio-temporal patterns at large temporal scales. After feature extraction, as 3D visual features already contain temporal information, methods generally focus on cross-modal alignment and may treat each frame as being independent during segmentation. Dynamic filtering and cross-modal attention are the two most common types of alignment strategies.

Dynamic filtering, which is sometimes referred to as dynamic convolution (DC) in the related literature [79, 81], is a special case of DC, which was proposed in [145] in the context of weather prediction and has been widely adopted in query-based object detection systems [146, 147]. Like a traditional convolution, DC convolves an input set of feature maps with the same kernel

across all locations. What is different in DC is that the kernel is composed of transformations of a conditioning feature vector, instead of being fixed model parameters. This allows the detection of user-defined patterns in the target set of feature maps. Gavrilyuk *et al*. [79] propose to instantiate the conditioning feature vector as the sentence feature vector, from which a kernel with one output channel is predicted via an MLP. This kernel is used as classification weights to output a foreground probability map for predicting the mask. Based on this idea, Wang *et al*. [81] make several improvements including the adoption of group convolution and pointwise convolution, as well as involving the target feature maps in the process of kernel generation (the math in the paper is poorly written and the illustrations are not very illustrative; please refer to the paper for details).

Among methods that rely on cross-modal attention mechanisms (similar to what we have described in Section 2.2) for feature alignment, Liu *et al*. [148] extend the multi-modal graph reasoning mechanism in [67] (introduced in Section 2.2) to the video domain by assigning an "action" attribute to words and extending attention computation and graph convolution [149] to the temporal dimension, while Wang *et al*. [82] devise a simple and effective bi-directional cross-modal attention network (sharing motivations with [66] and [128] introduced in Section 2.2), where in a language-guided vision attention module, visual and linguistic features are fused by element-wise multiplication followed by a layer of self-attention [69, 124], and in a vision-guided language attention module, pixel-wise visual features are used as queries and word-wise linguistic features are used as keys and values for gathering pixel-specific linguistic context.

Instead of relying on dynamic filtering or cross-modal attention, McIntosh *et al*. [150] propose a cross-modal alignment strategy based on capsules [151], a type of representation that specializes in encoding object parts and their relations. In their work, visual and linguistic features are encoded into capsules, and a capsule routing algorithm is proposed to fuse the two types of capsules into multi-modal capsules. A 3D convolutional segmentation decoder produces the

final segmentation masks for all video frames in one pass. Readers can refer to [151] and [152] for the details of capsules and their routing algorithms.

A general issue with the 3D convolutional video encoders is that they cause spatial misalignment in the encoded features [88, 153], which is detrimental to accurate spatial segmentation. The reason can be understood intuitively in the following way. First, the output of a 3D convolution kernel at a certain spatial location is aggregated from a fixed spatial neighborhood placed on multiple frames. As frames move, at each instance of time (frame), the same neighborhood contains different visual contents, and if we aggregate these contents over time, we mix them into something unintelligible (one extreme example would be that at one instance object 1 is at the location and on the next, object 2 appears at that location). Second, this problem can be aggravated by large temporal strides (which are necessary for action recognition to reduce redundancy in spatial details), as sparse temporal sampling can disrupt the temporal continuity of visual contents, which makes the example above more likely to happen.

To address this issue, Hui *et al*. [153] suggest to encode the input video with **both a video encoder and an image encoder**, where the two types of encoders produce complementary features: The video encoder encodes multiple frames at once and captures action information, while the image encoder encodes individual frames and captures fine spatial details. Linguistic features are integrated (*e.g*., via a cross-modal attention mechanism) with each of the two types of features, and the resulting cross-modal features are joined for segmentation. In general, the fact that temporal information modeling may be at odds with preserving fine spatial details in video recognition tasks is a more broadly recognized challenge, and readers may refer to the SlowFast Network [154] for more information.

Finally, very recently, several methods that exploit Transformers for learning cross-modal, cross-frame correspondences are proposed, which can work with both video and image encoders [88, 155]. Readers could refer to [156] for an introduction to the application of Transformers to the problem of video

instance segmentation, and [88, 155] for specific examples of the Transformers' application to the problem of VOSRE.

3

ANCHOR DIFFUSION FOR UNSUPERVISED VIDEO OBJECT SEGMENTATION

ABSTRACT

Unsupervised video object segmentation has often been tackled by methods based on recurrent neural networks and optical flow. Despite their complexity, these kinds of approaches tend to favor short-term temporal dependencies and are thus prone to accumulating inaccuracies, which cause drift over time. Moreover, simple (static) image segmentation models, alone, can perform competitively against these methods, which further suggests that the way temporal dependencies are modeled should be reconsidered. Motivated by these observations, in this paper we explore simple yet effective strategies to model long-term temporal dependencies. Inspired by the non-local operators of [124], we introduce a technique to establish dense correspondences between pixel embeddings of a reference "anchor" frame and the current one. This allows the learning of pairwise dependencies at arbitrarily long distances without conditioning on intermediate frames. Without online supervision, our approach can suppress the background and precisely segment the foreground object even in challenging scenarios, while maintaining consistent performance over time. With a mean IoU of 81.7%, our method ranks first on the DAVIS-16 leaderboard of unsupervised methods, while still being competitive against state-of-the-art online semi-supervised approaches. We further evaluate our method on the FBMS dataset and the ViSal video saliency dataset, showing results competitive with the state of the art.

3.1 INTRODUCTION

Video object segmentation (VOS) is a fundamental task in many important areas such as autonomous driving [33–35], robotic manipulation [36], video surveillance [37], and video editing [38]. Contemporary literature typically considers this problem in either the *semi-supervised* or the *unsupervised* setting. In both cases the objective is to predict, in every frame, pixel-level masks delineating certain objects of interest.

Under the semi-supervised setting, at test time methods can rely on a mask that specifies the object to segment. In contrast, the unsupervised setting does not provide any initialization. Without online supervision, the task might be considered ambiguous, as different objects could be considered of interest for different reasons, according to the application. Among researchers, the current consensus is to segment out foreground objects where a human gaze is more likely to focus [32], and for this reason, this task is considered to be more similar to semantic or saliency segmentation than instance segmentation. In more practical terms, an object is generally considered as *foreground* if it is sufficiently large, in motion and centered in the scene. In certain datasets (*e.g.*, FBMS [27] and ViSal [28]), in the same video, multiple foreground objects are considered, while in DAVIS-16 [16] only a single object is considered.

With the aim of tracking temporal changes of target objects, current state-of-the-art unsupervised approaches generally model motion cues in a video sequence via optical flow [40–47] or recurrent neural networks (RNNs) [39, 40]. Typically, these methods sequentially propagate features from the previous steps to the current one, thus making the current prediction depending on the entire history of the video.

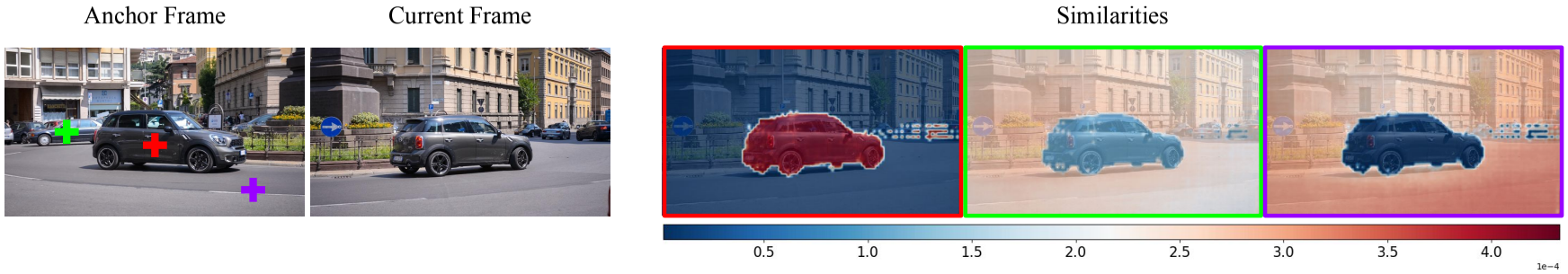


Figure 3.1: Examples of one-to-many similarities (right-hand side) learned by our method between single pixel embeddings belonging to the *anchor frame* to all the pixel embeddings from the *current frame* (left-hand side). The outlines of the dense similarities match the color of the corresponding pixel embedding in the anchor frame. Notice how the dense similarities with the pixel embedding from the foreground car (in red) produce a neat heat map that well identifies the object, while both sets of similarities with the pixel embedding from a "distractor" car (in green) and the road (in purple) are higher in correspondence of the background. These learned similarities are a simple and effective way of segmenting out foreground objects. Best viewed in colors.

Though having the potential of exploiting informative temporal cues, these approaches suffer from several limitations. RNNs often rely on training techniques such as truncated backpropagation through time to reduce the cost of parameter updates, which limits their long-term modeling capability [48]. Moreover, while LSTM's gating mechanism alleviates the issue of vanishing gradients [51, 52], the phenomenon of exploding gradients often requires clipping or rescaling the norm of the gradients during training [53]. Optical flow vectors only predict one-step motion cues at each frame in a video, which can accumulate errors over time. What is more, models relying on optical flow are typically trained on synthetic videos due to the high cost of per-frame and per-pixel labeling. Therefore, when applying these systems to real videos, the domain gap can cause the flow fields to contain several inaccuracies, especially when the foreground is nearly static [40].

In the video object segmentation community, the deterioration of performance over time in unsupervised VOS methods based on optical flow or RNNs is well known and has been widely discussed [44, 54–56]. For instance, Li *et al*. [44] demonstrate that, as a regular optical flow-based model progresses through frames, foreground embeddings become increasingly closer in feature space to the first frame's background as opposed to the foreground. Furthermore, Voigtlaender *et al*. [56] observe that a simple static segmentation model can achieve competitive results in the unsupervised VOS setting, which further corroborates the case for steering away from the sequential modeling strategies used by established methods.

Motivated by the above observations, in this work we opt for a much simpler solution, which is based on learning the similarity between pixels belonging to frames that can be arbitrarily far apart in time. To ensure representation consistency and reduce long-term drift, we propagate the features of the first frame (the "anchor") to the current frame via an aggregation technique inspired by the non-local operation introduced by Wang *et al*. [124]. This approach allows us to forgo sequential modeling, while at the same time enabling us

to deal with long-term dependencies and achieve high robustness over time, as shown in our experiments.

Despite its simplicity and online operability, our method outperforms the current state of the art [46] on the DAVIS-16 leaderboard by a margin of (absolute) 2.2% in terms of intersection-over-union, without resorting to auxiliary training data or post-processing. Moreover, it also achieves state-of-the-art results on FBMS [27] and the ViSal [28] video saliency benchmark. Code and pre-trained models are available at `https://github.com/yz93/anchor-diff-VOS`.

3.2 RELATED WORK

The problem of video object segmentation (VOS) is tackled by the computer vision community in the *unsupervised* or *semi-supervised* settings, which are defined by the level of supervision provided at test time.

Semi-supervised VOS methods are provided with a pixel-wise mask identifying the target object in the first frame of a video. When aiming at very high segmentation accuracy, methods generally perform online fine-tuning on the basis of this supervision [56, 92–97], sometimes exploiting data-augmentation techniques [92, 93] or self-supervision [56]. As online fine-tuning can take up to several minutes per video, many recently proposed methods renounce to it and instead aim at a faster online speed (*e.g.*, [54, 98, 103]). These faster semi-supervised approaches come in many flavors. For instance, Chen *et al*. [54] learn a metric space for pixel embeddings, which is then used to establish associations between pixels across frames, while Cheng *et al*. [98] suggest to individually track object parts from the first frame with a visual object tracker [102] and then aggregate them according to their similarity with the initialization mask.

Unsupervised VOS methods, instead, cannot rely on any supervision at test time and are often based on optical flow and RNNs. The purely optical flow-based MP-Net [47] discards appearance modeling and casts segmentation as

foreground motion prediction, an approach which poorly deals with static foreground objects. To address this problem, several methods (*e.g.*, LVO [40], SegFlow [41], MotAdapt [46] and MBN [45]) suggest to integrate appearance-based and optical flow-based features together, leading to variations of the "two-stream model" presenting two dedicated parallel branches. The drawbacks of these methods are threefold. First, flow estimation networks are typically trained on synthetic datasets and can thus result in poor performance when deployed in the real world. Second, while modeling long-term temporal dependencies is critical for adapting to significant online changes, the vector fields can only model short-term one-step dependencies. Targeting this issue, Tokmakow *et al*. [40] proposed to extend the horizon spanned by optical flow-based features by employing a convolutional gated recurrent unit [50]. Third, vector fields cannot distinguish foreground and background objects when they move in a synchronized fashion (*e.g.*, the cars in a traffic jam). Li *et al*. [45] attempt to address this issue by employing a bilateral network for detecting the motion of background objects. Our investigations with a much simpler appearance-based approach show that optical flow may not be an essential component of unsupervised VOS systems.

RNN-based models are often challenged by the problems of exploding and vanishing gradients [51, 52], which limit their long-term modeling capability. Among the methods that make use of recurrent connections, Song *et al*. [39] propose a novel convolutional long short-term memory [49] architecture, in which two atrous convolution [104] layers are stacked along the forward axis and propagate features in opposite directions.

Recently, it has been shown [44, 55, 56] that both recurrent and optical flow-based methods significantly suffer from a deterioration in the quality of their predictions over time. This has motivated the several approaches (including ours) that tackle video object segmentation by simply learning similarities between pixel embeddings (*e.g.*, [44, 45, 54, 105]). These methods first select a set of seed pixels that are most likely to belong to the foreground object and then

classify all other pixels based on their similarities to these seeds, for instance by thresholding or by propagating labels between neighbors. Fathi *et al*. [105] adopt this approach for semantic instance segmentation, in which the pairwise pixel similarity function measures the likelihood of two pixels belonging to the same instance. IET [44] extends this concept to video sequences. Similarly, it selects a set of foreground and background seeds for each frame and organizes them into tracks. It then segments each frame individually based on pixel similarities with the foreground and background seeds. Note that IET utilizes pre-trained instance embeddings. MBN [45] extends IET with a bilateral filtering network that filters false-positive foreground predictions using optical flow features and an energy minimization procedure on a graph of seeds sampled from a few consecutive frames. When segmenting frame t, MBN classifies each pixel by assigning it the label of the seed (sampled from frames $t-1$, t, and $t+1$) with which it has the smallest embedding distance.

The main drawback of these methods is the complexity involved in the procedures of seed selection, ranking, and classification, critical for achieving good performance. Moreover, these algorithms also depend on multiple scores such as motion saliency and objectness that need to be carefully calibrated and combined into one final metric.

Albeit our proposal is related to this last class of approaches, it is considerably simpler. Instead of separately learning individual components from image datasets and classifying pixels based on similarities with seeds, our method performs similarity learning, feature propagation, and binary segmentation in a single network.

3.3 METHOD

We are interested in the task of binary segmentation of a sequence of video frames, where the final performance is measured by the average segmentation quality of individual frames. Therefore, our method should perform well under

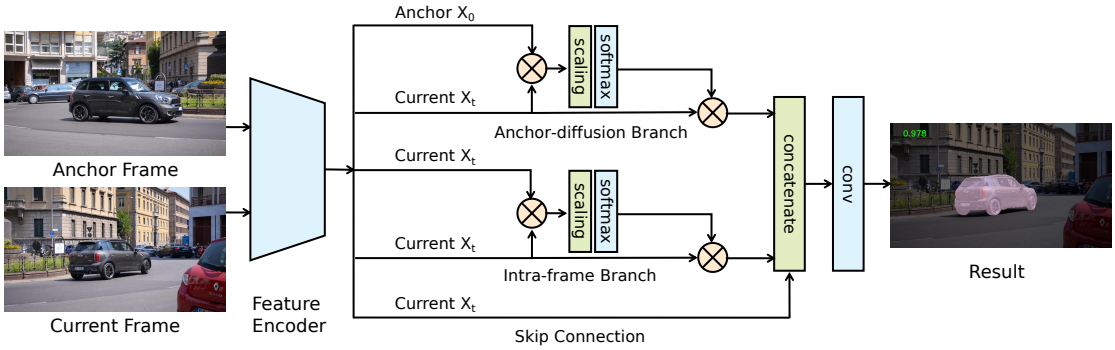


Figure 3.2: Overall pipeline of the proposed method.In the anchor-diffusion branch, pixel embeddings in the current frame are linearly transformed by similarity scores with pixel embeddings in the anchor frame, and concatenated with both the output of the intra-frame branch and the output of the skip connection.

two aspects. First, similarly to what is expected from a static segmentation model, it should be able to provide accurate segmentation masks of foreground objects in individual frames. Second, it should be able to well adapt to the appearance changes of the foreground objects throughout the whole video.

In the proposed *anchor diffusion network* (AD-Net) (Figure 3.2), we address both requirements in a single model trained end-to-end by leveraging the recently proposed non-local operations of Wang *et al*. [124]. Closely related to the concept of self-attention [69], a non-local operation is a neural network building block that captures the dependencies within a set of input feature vectors.

To achieve our first goal, a non-local operation is applied to the encoding of the target frame, in a similar way it is applied for semantic image segmentation [157], forming the *intra-frame branch* of our overall model. To achieve our second goal, we propagate information between two frames: a fixed anchor frame and the current frame, forming the *anchor-diffusion branch* of our overall model. We name the branch this way to give relevance to its functionality of "diffusing" information from the anchor to the large number of target frames at test time, which encourages foreground embeddings of each target frame to be consistent over time.

In the following, we describe our pipeline in more detail.

Pipeline. The input of our model consists of a pair of images: a fixed anchor frame I_0 and the frame to segment I_t. The overall pipeline is schematically illustrated in Figure 3.2. First, a feature encoder (the fully-convolutional DeepLabv3 [104]) encodes I_0 and I_t into the corresponding embeddings $X_0 \in \mathbb{R}^{hw \times c}$ and $X_t \in \mathbb{R}^{hw \times c}$, where c denotes the number of channels and h, w denote the height and width of the frame. We refer to the c-dimensional feature vector at each location as a *pixel embedding*. The output of this first stage is then fed to three parallel branches: a skip connection with an identity mapping [112], the intra-frame branch, and the anchor-diffusion branch. X_t is fed to all branches, while X_0 only to the anchor-diffusion branch. Finally, the resulting features from the three branches are concatenated together along the channel dimension before the classification layer.

The entire network is trained end-to-end with a binary cross-entropy loss. Though any frame could be selected as the anchor frame, in practice we always choose the first frame for computational convenience and because, in benchmarks, the first frame is guaranteed to contain the foreground objects. During training, the first frame and a random frame are sampled from the video.

3.3.1 *Anchor diffusion*

As described earlier, X_0 and X_t represent the embeddings of the anchor and the current frame respectively. To reinforce the foreground signal, it is important to know which pixel embeddings in X_t correspond to the background introduced throughout a video. To achieve this, in the anchor-diffusion branch we compute a transition matrix $P \in \mathbb{R}^{hw \times hw}$ which establishes dense correspondences between each pair of pixels from X_0 and X_t and use it to map X_t to a new encoding $\tilde{X}_t$, in which the pixel embeddings are weighted according to their similarity with the foreground:

$$\tilde{X}_t = PX_t. \tag{3.1}$$

As qualitatively illustrated in Figure 3.1 and Section 3.6.4, this procedure significantly strengthens the foreground while weakening the background. It is worth noting that one can also simply use the concatenation of X_0 and X_t to achieve this goal. However, we find in our experiments that the correspondence learning in Equation (3.1) can better localise the foreground objects.

Similarly to [124], the transition matrix is defined as

$$P = \text{softmax}(\frac{1}{z} X_0 X_t^T), \tag{3.2}$$

where $X_0 X_t^T$ is a pairwise dot product similarity between each pair of pixel embeddings in X_0 and X_t. Following [69, 158], we scale the dot product with a factor $z = \sqrt{c}$, where c is the number of channels of X_0 and X_t. The rationale is that, for embeddings with high dimensionality, dot products can be very large and thus push the output of the softmax to regions where gradients are small [69]. The softmax function normalizes each row of $\frac{1}{z} X_0 X_t^T$ to sum to one, thereby preserving scale invariance of the pixel embeddings. Without normalization, multiplying $\frac{1}{z} X_0 X_t^T$ with X_t can entirely change the scale of the pixel embeddings.

In the case of the intra-frame branch, each output pixel embedding can be considered as a global aggregation of all input pixel embeddings weighted by pairwise appearance similarity. It has been shown that such use of non-local operations [124] can harness long-range spatial information, which is beneficial for semantic segmentation [157]. Empirically, as detailed in the ablation studies of Table 3.1, we found that incorporating this branch in addition to the anchor-diffusion branch further improves the performance of the model.

The intra-frame branch improves segmentation accuracy but does not address the temporal changes in a video sequence. Conversely, the anchor-diffusion branch models pairwise dependencies between frames, with the result of enhancing the consistency of pixel embeddings and reducing drift.

Qualitative analysis. As shown in Figure 3.1, each of the colored pixels in the anchor frame finds desirable correspondences in the current frame. The

foreground car pixel embedding (red) has high similarity with pixel embeddings of the foreground car in the current frame despite the appearance change and sets off a neat contrast with the background that precisely outlines the target object. Conversely, both heat maps of the distractor car pixel embedding (green) and the road pixel embedding (purple) have higher similarity values in the background region of the current frame, which is what is expected for pixel embeddings of the background class. Moreover, as the distractor car is not present in the current frame, its pixel embedding only finds weak and widespread correspondences in the general background region, with a weak separation between the foreground and the background. In contrast, the pixel embedding corresponding to the asphalt, which represents the common material appearing in both frames, shows a higher similarity with the road region and sets a larger separation between the foreground and the background. More qualitative results illustrating the similarity between pixel embeddings are shown in Section 3.6.4.

Overall, these results show that the transition matrix P learns a similarity metric that can well identify common objects/materials across two frames. Therefore, when used in Equation (3.1), P can strengthen the signal from pixels which have strong correspondences in the anchor frame and weaken the signal from pixels which do not. As the foreground target object is almost always present in both frames while the background changes relatively quickly, our diffusion process generally strengthens the foreground and suppresses the background.

In Figure 3.3, instead, we report how foreground embeddings change over time by computing the average cosine distance between the foreground embeddings of a later frame and those of the first frame. Notice how the embeddings of the baseline quickly grow apart, while the ones learned with our proposed method are significantly stabler. This suggests that AD-Net is capable of preserving the foreground information from the first frame in a video over a long time period.

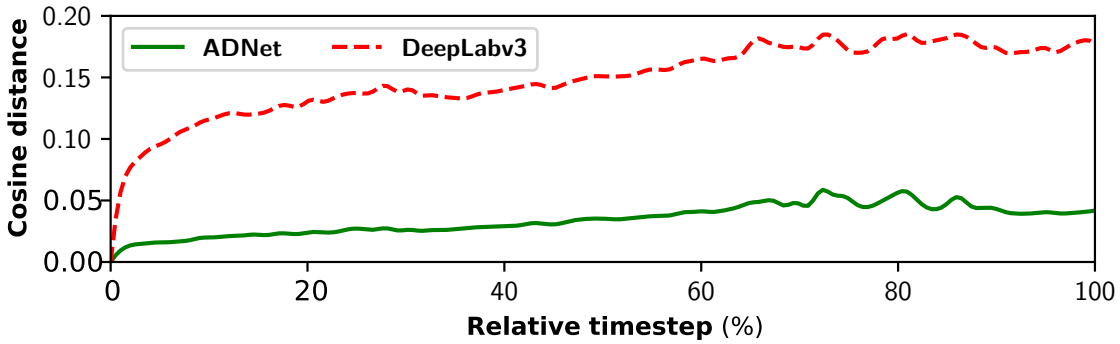


Figure 3.3: Temporal consistency of pixel embeddings over time, measured as cosine distance between foreground pixel embeddings in the anchor frame and foreground pixel embeddings in progressively more distant frames.

3.4 EXPERIMENTS

In the following, after discussing important implementation details regarding our architecture and training procedure, in Section 3.4.1 we illustrate the three benchmarks we adopted, in Section 3.4.2 we describe several ablation studies and in Section 3.4.3, we provide an extensive comparison with the state of the art.

Implementation details. We employ the fully-convolutional DeepLabv3 [104] as the feature encoder, and initialize its ResNet101 [112] backbone with weights pre-trained on ImageNet. The other layers in DeepLabv3 are randomly initialized. The configuration of the dilation rates follows the original model [104] and presents a total stride of 8. We modify the number of output channels in the last layer to 128, which corresponds to c in Section 3.3.

In the anchor-diffusion step, the spatial dimensions of each image encoding are flattened and transposed where appropriate in order to perform batched matrix multiplication. The outputs of the three branches are concatenated along the channel axis and reduced to dimension 128 via a 1×1 convolution with LeakyReLU non-linearity and dropout rate 0.1. The final classification layer is implemented as a 1×1 convolution with a single output channel followed by a sigmoid layer.

Training. Each training example consists of a pair of images. Given a randomly sampled video, we use the first frame as the anchor image and a randomly

sampled frame as the second image. We also experimented with randomly sampling both frames and observed slightly worse performance. Each input frame is cropped to a randomly-sized region enclosing the ground-truth foreground. Random rotations are performed at 45-degree increments, with a probability of 51% of not rotating and equal probabilities of rotating to any of the remaining angles.

The model is trained with binary cross-entropy loss. Network parameters are optimized via stochastic gradient descent with a weight decay of 0.0005. The initial learning rate is set to 0.005 and follows a "poly" adjustment policy [104], where the initial learning rate is multiplied by $(1 - \frac{iter}{40{,}000})^{0.9}$ at each iteration. The model is trained for 30,000 iterations with batch size 8. Raw predictions are upsampled via bilinear interpolation to the size of the ground-truth masks.

Inference. At test time, the features of the anchor frame are computed once and reused throughout the video. Multi-scale and mirrored inputs are employed to enhance the final performance. Each input image is scaled by factors of 0.75, 1.00, and 1.50 and horizontally flipped. The final heatmap is the mean of all output heatmaps. Thresholding at 0.5 produces the final binary labels.

Instance pruning. Since semantic segmentation approaches like the one we use lack the notion of instance and some videos from the DAVIS-16 dataset [16] present multiple objects that can be deemed as foreground, we experiment with a simple set of post-processing steps to prune "non-foreground" objects. As instance trajectories measure the spatial changes of an instance, they can be used to detect background instances which have distinct trajectory patterns from the foreground instance. First, we establish online temporal correspondences by using a pre-trained object detection model [159] to predict the locations of all objects and track the trajectory of each detection across the entire video using an intersection-over-union criterion between consecutive bounding boxes. Once object tracks have been established, we use the cumulative area of instance masks across frames as a proxy to identify foreground objects, thus pruning small objects or objects that are only present in a fraction of the video. This process

produces a filtering mask, which is multiplied element-wise with AD-Net predictions to obtain the final predictions. More details and hyper-parameters related to this process (which we refer to as *instance pruning*) are provided in Section 3.6.

3.4.1 *Benchmarks*

Datasets. DAVIS [16] is a benchmark and yearly challenge for video object segmentation (VOS). Unsupervised methods are trained and evaluated with the DAVIS-16 dataset, which annotates a single foreground entity. There are 30 videos for training and 20 videos for validation. We train our method on the training set and evaluate on the validation set.

The FBMS [27] dataset is another challenging benchmark for unsupervised video object segmentation containing 29 training videos and 30 test videos. Following [39, 40, 44–46], we evaluate on the test set.

Finally, ViSal [28] is a video salient object detection dataset containing 17 video sequences. Despite our method has not been designed for the task of saliency, we can easily report results on this benchmark, too.

Evaluation metrics. For DAVIS, we adopt the official evaluation metrics of mean region similarity $\mathcal{J}$, which is the intersection-over-union of the prediction and ground truth, and mean contour accuracy $\mathcal{F}$, which is the F-measure defined on contour points from the prediction and the ground truth. To provide more insights, we plot precision-recall (PR) curves on all three benchmark datasets. On the FBMS dataset, the main evaluation metric is the F-measure. On the ViSal dataset, we report the mean absolute error (MAE) and the F-measure. For definitions of MAE and the F-measure, we refer readers to [160].

Model	$\mathcal{J}$(%)	$\triangle_{\mathcal{J}}$	$\mathcal{F}$(%)	$\triangle_{\mathcal{F}}$
Baseline [104]	75.41	0.00	75.58	0.00
Baseline + intra-frame	76.17	+0.76	75.38	-0.20
Baseline + anchor	76.84	+1.43	75.76	+0.18
Baseline + anchor-diffusion	77.43	+2.02	76.78	+1.20
AD-Net (single scale)	78.26	+2.85	77.11	+1.53

Table 3.1: Ablation study conducted using the DAVIS-16 validation set. $\triangle_{\mathcal{J}}$ and $\triangle_{\mathcal{F}}$ denote, respectively, absolute improvements in region similarity and contour accuracy.

3.4.2 *Ablation studies*

We conduct several ablations to evaluate the effectiveness of the anchor-diffusion procedure. First, we evaluate DeepLabv3 [18] *as-is*, simply fine-tuning it on the DAVIS training set. This semantic segmentation baseline (designed for static images) performs on par with some state-of-the-art unsupervised VOS methods (see Table 3.2). This is in line with what is described by Voigtlaender *et al*. [56], but it is rather curious that it still applies after two years of progress. Clearly, the competitive performance can be partially attributed to the high performance of DeepLabv3 for the similar task of semantic segmentation of static images. However, this result also shows that existing unsupervised VOS techniques are not able to successfully model and leverage temporal dependencies and that different approaches should be sought.

Starting from this baseline, we evaluate four variants that differ in the embeddings they consider at the terminal concatenation layer (see Figure 3.2). Each corresponds to a row below *Baseline* in Table 3.1. The first variant ("intra-frame") computes non-local features within the same frame X_t and without the anchor-diffusion branch. The second ("anchor") simply concatenates X_0 to X_t. The third performs anchor diffusion on X_0 and X_t, and concatenates the results with X_t, without features from the intra-frame branch. The fourth (our final model, AD-Net) concatenates both the output of the intra-frame branch and that of the anchor-diffusion branch with X_t.

The "intra-frame" variant improves over the baseline, which shows the potential of utilizing context information within the current frame. The "anchor" variant demonstrates the general usefulness of an anchor frame, despite the apparent limitation that the fixed representation of the anchor frame does not adapt to changes in the current frame. The solid performance gains validate our motivation to further develop the anchor-diffusion mechanism. The "anchor-diffusion" variant illustrates the efficacy of the proposed feature diffusion mechanism across the anchor and current frames. It brings a performance boost of 2.02 (absolute) points over the baseline, larger than the contribution brought by the "intra-frame" and "anchor" variants.

3.4.3 *Comparison with the state of the art*

In Table 3.2, we evaluate AD-Net against state-of-the-art unsupervised VOS methods on the DAVIS public leader-board and also provide the performance of several popular semi-supervised methods as a term of reference. AD-Net attains the highest performance among all unsupervised methods on the DAVIS validation set, while also performing very competitively on the FBMS test set. In particular, on DAVIS we outperform the second-best method (MotAdapt [46]) by an absolute margin of 2.2% in $\mathcal{J}$ and 0.8% in $\mathcal{F}$ before applying the post-processing step of instance pruning. After applying instance pruning as described earlier, AD-Net achieves the final performance of 81.7 in $\mathcal{J}$ and 80.5 in $\mathcal{F}$, leading the second-best method by 4.5 and 3.1 absolute points respectively. Also, despite being unsupervised at inference time, AD-Net outperforms many semi-supervised methods which instead require to be initialized with a mask in the first frame.

After our proposed AD-Net, the second and third-best ranking methods are MotAdapt [46] and PDB [39], which are particularly representative of two classes of methods.

	Method	FF	OF	CRF	DAVIS $\mathcal{J}$	DAVIS $\mathcal{F}$	FBMS F-measure
Semi.	PReMVOS [95]	✓	✓	✓	84.9	88.6	-
	OSVOS [92]	✓			79.8	80.6	-
	MSK [97]	✓	✓	✓	79.7	75.4	-
	PML [54]	✓			75.5	79.3	-
	SFL [41]	✓	✓		76.1	76.0	-
	VPN [161]	✓	✓		70.2	65.5	-
Unsupervised	FST [162]		✓		55.8	51.1	69.2
	ELM [163]		✓		61.8	61.2	-
	SFL [41]		✓		67.4	66.7	-
	LMP [47]		✓	✓	70.0	65.9	77.5
	FSEG [42]		✓		70.7	65.3	-
	LVO [40]		✓	✓	75.9	72.1	77.8
	ARP [43]		✓		76.2	70.6	
	PDB [39]			✓	77.2	74.5	**81.5**
	MotAdapt [46]		✓	✓	77.2	77.4	79.0
	AD-Net (multiple scales)				**79.4**	**78.2**	81.2
	AD-Net + I.Prun. (ours)				**81.7**	**80.5**	-

Table 3.2: Performance on DAVIS-16 validation set. FF: first-frame annotations; OF: optical flow; CRF: random conditional field.

PDB is representative of top-performing RNN-based methods. Although, in theory, RNNs could model long-range time dependencies, in practice they are constrained to model relatively short sequences. First, as the computational graph of an (unrolled) RNN grows in depth with the length of a video sequence, backpropagation is typically limited to a few time steps (*e.g.*, 5 in RGMP [55]). Such backpropagation cannot guarantee long-term dependency modeling [48]. Second, despite the gating and memory mechanisms adopted by LSTMs and GRUs, long propagation paths of gradients still cause exploding or vanishing gradients [51, 52].

Conversely, MotAdapt is representative of top-performing methods that employ optical flow. It consists of a two-stream architecture, which dedicates two network branches (trained jointly but with different parameters) to process RGB images and pre-computed optical flow fields. The two-branch network is further fine-tuned at inference time, with pseudo-labels generated by a teacher network. Although optical flow is an intuitive way to model inter-frame dependencies

and aid segmentation, results in Tables 3.1 and 3.2 demonstrate that simply developing a better appearance-based model can overshadow the benefits of a dedicated optical flow branch. Moreover, the strategy of fine-tuning at inference time adopted by MotAdapt and many semi-supervised methods is a time-consuming process, taking many seconds up to minutes per video. In contrast, AD-Net leverages a simpler architecture, which makes it fast at inference time. Without instance pruning, it runs online and at 4 frames per second on an NVIDIA TITAN X GPU, with frames at the original DAVIS resolution of 854×480. Speed can be easily traded off at a small cost in performance, by using frames with lower resolution and/or a lighter architecture.

The precision-recall analysis of AD-Net is presented in Figure 3.4, where we demonstrate that our approach generally outperforms also existing salient object detection methods. AD-Net achieves superior performance in all regions of the PR curve on the DAVIS validation set, maintaining significantly higher precision at all recall thresholds. On the challenging FBMS test set, AD-Net maintains a clear advantage below the 90% recall threshold. On the ViSal dataset, it is noteworthy that nearly perfect precision is maintained up until the 60% recall rate, which is higher than the other methods.

Evaluation as video saliency. The definition of salient objects in a video for benchmarks like ViSal [28] is very related to the one of "foreground objects" for benchmarks like DAVIS or FBMS (see Section 3.1). Annotations in salient object detection datasets can vary from coarse annotations such as bounding boxes to fine-grained pixel-level real-valued scores, and sometimes even take the form of human eye fixations. ViSal provides pixel-level annotations as binary labels, annotating large, moving objects as the foreground and everything else as the background. Despite the many types of annotations, evaluation metrics are fairly standard and use pixel-level annotations either in a binarized form (PR curve and F-measure) or as normalized saliency scores between 0 and 1 (MAE), which are directly applicable to the scores produced by AD-Net.

	Saliency Methods	DAVIS MAE ↓	DAVIS F ↑	FBMS MAE ↓	FBMS F ↑	ViSal MAE ↓	ViSal F ↑
Image	Amulet [164]	0.082	69.9	0.110	72.5	0.032	89.4
	SRM [165]	0.039	77.9	0.071	77.6	0.028	89.0
	UCF [166]	0.107	71.6	0.147	67.9	0.068	87.0
	DSS [160]	0.062	71.7	0.083	76.4	0.028	90.6
	MSR [167]	0.057	74.6	0.064	78.7	0.031	90.1
	NLDF [168]	0.056	72.3	0.092	73.6	**0.023**	**91.6**
	DCL [169]	0.070	63.1	0.089	72.6	0.035	86.9
	DHS [170]	**0.039**	75.8	0.083	74.3	0.025	91.1
	ELD [171]	0.070	68.8	0.103	71.9	0.038	89.0
	KSR [172]	0.077	60.1	0.101	64.9	0.063	82.6
	RFCN [173]	0.065	71.0	0.105	73.6	0.043	88.8
Video	FGRNE [174]	0.043	78.6	0.083	77.9	0.040	85.0
	FCNS [175]	0.053	72.9	0.100	73.5	0.041	87.7
	SGSP [176]	0.128	67.7	0.171	57.1	0.172	64.8
	GAFL [28]	0.091	57.8	0.150	55.1	0.099	72.6
	SAGE [177]	0.105	47.9	0.142	58.1	0.096	73.4
	STUW [178]	0.098	69.2	0.143	52.8	0.132	67.1
	SP [179]	0.130	60.1	0.161	53.8	0.126	73.1
	AD-Net (ours)	0.044	**80.8**	**0.064**	**81.2**	0.030	90.4

Table 3.3: Salient object detection performance of AD-Net, compared against 18 popular saliency prediction methods.

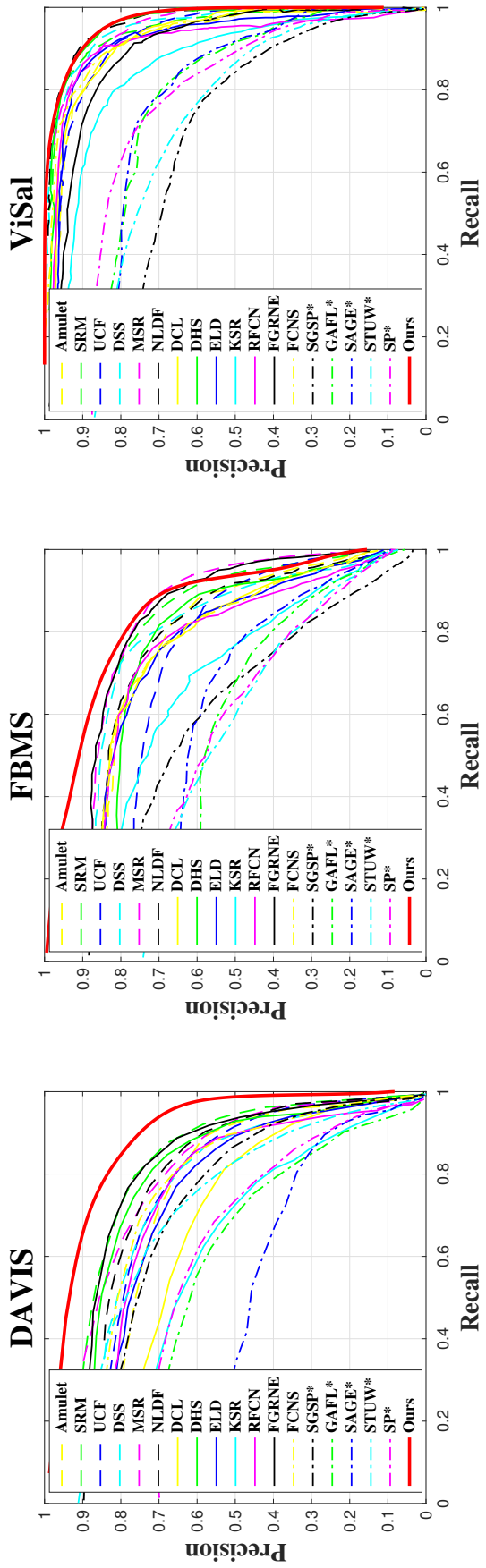


Figure 3.4: AD-Net results with PR curves on the DAVIS, FBMS, and ViSal datasets.

As shown in Table 3.3, the proposed AD-Net improves the state of the art for both DAVIS and FBMS also for standard saliency scores, showing consistency with Table 3.2. The largest improvements lie in FBMS, where both MAE and F-measure significantly outperform previous records. On DAVIS, F-measure is the highest among all methods with a significant leading margin. On the ViSal dataset, AD-Net achieves the best MAE (lower is better) among all video saliency models and obtains F-measure close to the overall best method. Remarkably, despite not having trained for the task of saliency prediction, we outperform previous saliency methods under saliency metrics on DAVIS and FBMS, and achieve very competitive results on ViSal.

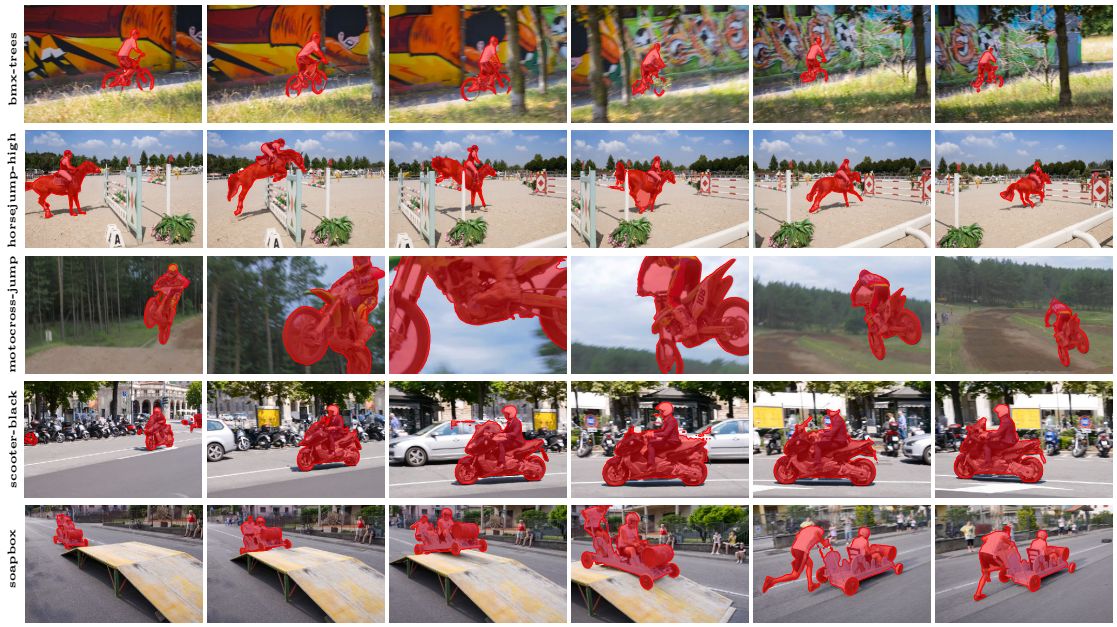


Figure 3.5: Segmentation results on DAVIS-16 validation set videos, obtained using our model without any online fine-tuning.

3.5 DISCUSSION

In this paper, we proposed Anchor Diffusion Network (AD-Net), a method for unsupervised video object segmentation based on non-local operations. Instead of modeling temporal dependencies with recurrent connections or

adopting pre-computed optical flow like contemporary work, we argue for a significantly simpler and more effective approach, which consists in establishing correspondences of pixel embeddings between a reference frame and the current one. With this strategy, we can easily model long-term temporal dependencies at a low computational cost. We show how, during inference, this procedure is able to suppress the background while preserving the foreground even when abrupt changes in appearance occur. Quantitative evaluations across three standard benchmarks demonstrate the advantage of our proposed method on the task of unsupervised video object segmentation with respect to the state of the art. Moreover, our method is also surprisingly competitive against the state of the art in semi-supervised video object segmentation and video saliency.

The cases where our model demonstrates a significant advantage with respect to static models are frames where the object's appearance deviates considerably from the "norm," *i.e.*, how the object usually looks. This is because our model learns a similarity function that adapts to such changes by design—during training, it has to learn to associate pixels that belong to the same object, even when they are somewhat dissimilar. By doing this, the similarity function learns to be invariant to non-essential changes in the object. Nonetheless, our approach has several limitations. The first is that such ability for adaptation is not unlimited—there are cases when the changes are too big for the similarity function to handle, and this is hard to solve without scaling up the training data. Second, while having the potential to be extended to the semi-supervised setting, *i.e.*, dealing with multiple objects, our algorithm is not designed for that setting and its efficacy is unknown in that territory. Third, our algorithm struggles in some videos when the target object is too similar to objects in the background while all of them can be considered salient. In those cases, optical flow could probably come in useful as additional cues for telling the objects apart. But this is an inherent limitation due to the ambiguity of the problem setup.

Having discussed the long-term efficacy of AD-Net in Sections 3.1 and 3.3 and just a little more on its adaptability to online appearance changes, one

question stands out: Are there better choices for the reference frame than the first frame? We outline an approach to answering this question in the following. First, assuming our mechanism stays the same, *i.e.*, using a single reference frame without a memory module, then we should investigate the choices of (1) the middle and the last frames (sacrificing online operability in this case), and (2) the frame before the current frame at a certain fixed distance, *e.g.*, 1, 5, 10 or 15 frames apart. Second, we might consider the possibility of adding a memory module, which stores a certain number of past frames and predictions as the model makes causal predictions. In this setting, the core operations of "diffusing" information from history frames to the current frame, *i.e.*, Equations (3.1) and (3.2), can be applied individually on each frame in the memory and the current frame. We could also make use of the past predictions in addition to features, for instance, via concatenation or mask pooling. The main hyperparameter to consider in this approach is the number of past frames to store. In the literature on the broader topic of video object segmentation, *e.g.*, [180] and [181], the general observation is that incorporating longer history context provides more benefits, and techniques such as downsampling and hierarchical memory are effective ways to reduce GPU memory consumption while maintaining the model efficacy. Finally, taking a learning approach towards the selection of the reference frame is another interesting direction, for instance, as having been done in [182] for semi-supervised video object segmentation and [183] for interactive video object segmentation.

The last point we'd like to make in this chapter before entering the appendices section is the inspiration brought by the overwhelming success of the Transformer [69] model in various related tasks recently. These are the things we would do differently based on the new observations.

The first concerns scaling up the model (more attention layers at more levels of the network) and training data, which is currently in effect prohibited by the GPU memory consumption of the attention block in AD-Net, being quadratic to the number of pixels in a frame. As we shall see in Chapters 4 and 5, scaling up

attention and making use of multiple levels of visual features are both important aspects in designing a state-of-the-art segmentation model. And as mentioned before, the efficacy of the learned similarity function crucially depends on the amount of training data. One very recent insight we could draw on is the use of locality in a multi-layer attention network, as the representative work of the Swin Transformer [78, 184] has demonstrated. Essentially, by confining the range of pair-wise attention to local windows and shifting window positions across layers, we can harvest the global correlation modeling power of fully-connected attention layers while maintaining a computational cost linear to the number of pixels at the same time. The second insight is the mechanism of learnable embeddings as "queries." In more recent object detection systems (*e.g.*, [75, 76, 146, 147]), a set of learnable embeddings serve as latent object features, which essentially encode learned positional and semantic priors after training is done. These embeddings are model parameters and are optimized the same way as the other model parameters. Some methods (*e.g.*, [146, 147]) predict dynamic weights from these queries (for localizing and classifying objects), and others (*e.g.*, [75, 76]) use them literally as "queries" in a Transformer (see Section 2.2 for the concepts of *key*, *value*, and *query* in a Transformer). What we could explore in anchor diffusion is to use learnable embeddings to query the reference frame in order to gather context, and then append this context to features of the current frame. By replacing the pixel-level, pair-wise attention in AD-Net with attention based on a fixed number of learnable embeddings, we could dramatically reduce the amount of compute. One potential issue with this approach is that it is known that these learnable embeddings take an enormous amount of data to learn (*i.e.*, convergence is slow).

The second improvement (or investigation) that we could try is to replace the attention blocks in AD-Net with standard Transformer layers. This amounts to changing the anchor-diffusion branch to a Transformer decoder layer (having both self-attention and cross-attention) and the intra-frame branch to a Transformer encoder layer (having only self-attention). There are several differences

between our implementation and a standard Transformer layer (encoder or decoder), including non-linear projections with layer normalization [185], residual connections, and a feed-forward sub-layer. It will be interesting to see if adopting these more powerful constructs will bring any performance enhancements. Moreover, we could try stacking up layers (and this goes back to the point of scaling up)—modern Transformer-based architectures use multiple Transformer layers instead of a single layer. This is a promising direction to go especially when we can apply some of the compute-saving techniques described earlier.

3.6 APPENDICES

3.6.1 *Global comparison*

Table 3.4 includes all metrics reported in the official DAVIS 2016 benchmark [16]. Our method outperforms competing methods in the main evaluation metrics of mean region similarity $\mathcal{J}$ and mean contour accuracy $\mathcal{F}$. The small decay measure for both $\mathcal{J}$ and $\mathcal{F}$ demonstrates AD-Net's long-term efficacy.

3.6.2 *Per-sequence comparison*

Figures 3.6 and 3.7 compare the per-sequence $\mathcal{J}$ and $\mathcal{F}$ of AD-Net against the top seven competing methods on the leaderboard. Our method performs well on videos presenting a variety of challenges, such as appearance change (Car-Shadow, Parkour), cluttered background (Car-Roundabout, Scooter-Black), occlusion (Libby, Bmx-Trees), fast motion (Bmx-Trees, Dog, Parkour), *etc*.

Measure	ADNet	MotAdapt[46]	PDB[39]	ARP[43]	LVO[40]	FSEG[42]	LMP[47]	SFL[41]	TIS[186]	ELM[163]	FST[162]	CUT[187]	NLC[188]	MSG[189]	KEY[190]	CVOS[191]	TRC[192]
$\mathcal{J}$ Mean ↑	**81.7**	77.2	77.2	76.2	75.9	70.7	70.0	67.4	62.6	61.8	55.8	55.2	55.1	53.3	49.8	48.2	47.3
$\mathcal{J}$ Recall ↑	90.9	87.8	90.1	**91.1**	89.1	83.5	85.0	81.4	80.3	67.2	64.9	57.5	55.8	61.6	59.1	54.0	49.3
$\mathcal{J}$ Decay ↓	2.2	5.0	0.9	7.0	**0.0**	1.5	1.3	6.2	7.1	9.8	**0.0**	2.2	12.6	2.4	14.1	10.5	8.3
$\mathcal{F}$ Mean ↑	**80.5**	77.4	74.5	70.6	72.1	65.3	65.9	66.7	59.6	61.2	51.1	55.2	52.3	50.8	42.7	44.7	44.1
$\mathcal{F}$ Recall ↑	**85.1**	84.4	84.4	83.5	83.4	73.8	79.2	77.1	74.5	65.4	51.6	61.0	51.9	60.0	37.5	52.6	43.6
$\mathcal{F}$ Decay ↓	0.6	3.3	**-0.2**	7.9	1.3	1.8	2.5	5.1	6.4	8.8	2.9	3.4	11.4	5.1	10.6	11.7	12.9
$\mathcal{T}$ (GT 8.8) ↓	36.9	27.9	29.1	39.3	26.5	32.8	57.2	28.2	33.6	25.1	36.6	27.7	42.5	30.1	26.9	**25.0**	39.1

Table 3.4: Detailed evaluation results on the DAVIS 2016 validation set. We analyze region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, and temporal stability $\mathcal{T}$ in terms of mean, recall, and decay, and compare with state-of-the-art methods from the DAVIS 2016 leaderboard.

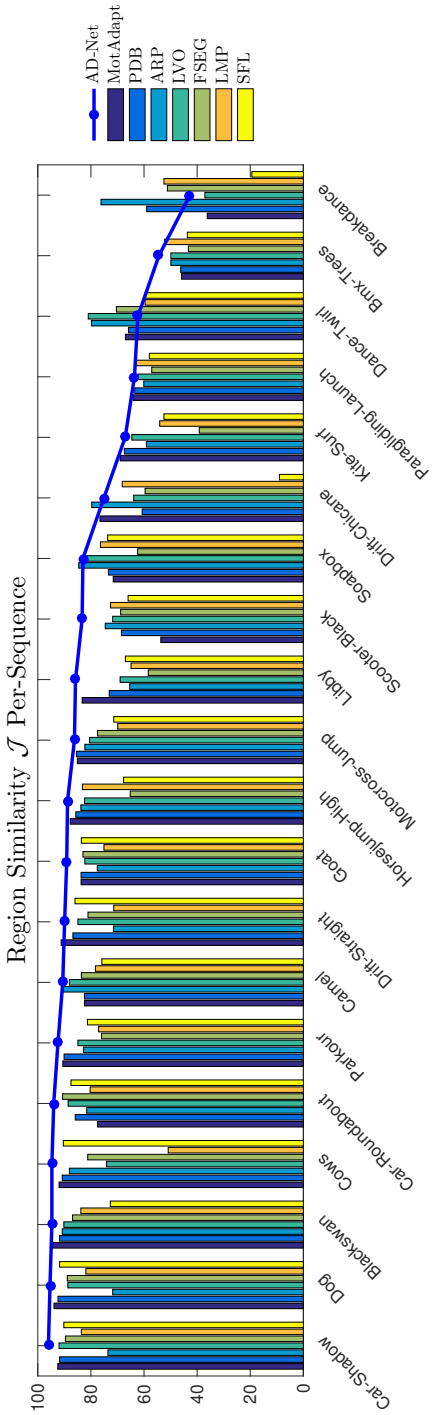


Figure 3.6: Per-sequence results of mean region similarity $\mathcal{J}$ against top 7 methods on the public leaderboard of DAVIS 2016. The blue line indicates AD-Net, while bars represent other methods. Sequences are organised in descending order of the performance of our method.

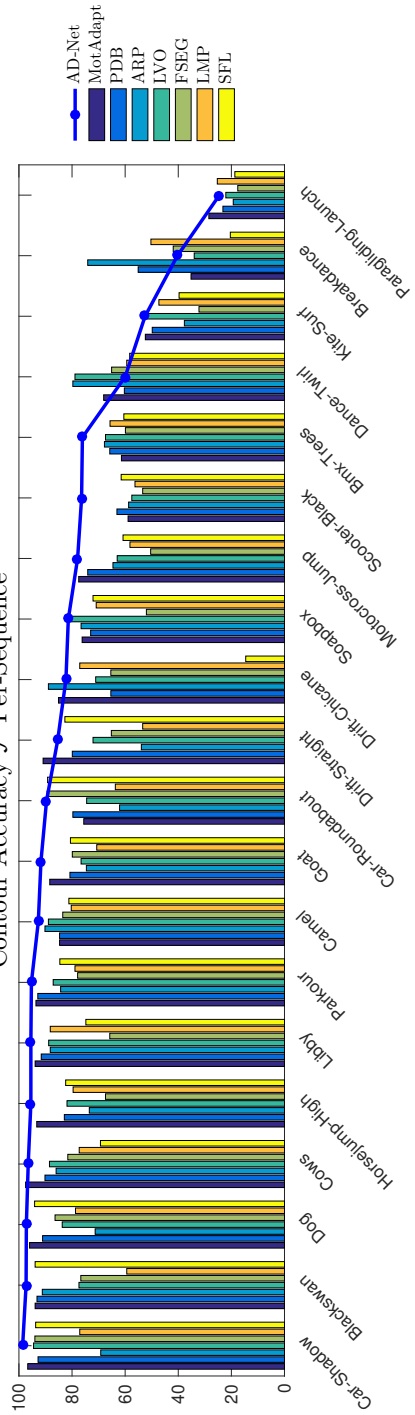


Figure 3.7: Per-sequence results of mean contour accuracy $\mathcal{F}$ against top 7 methods on the public leaderboard of DAVIS 2016. The blue line indicates AD-Net, while bars represent other methods. Sequences are organised in descending order of the performance of our method.



Figure 3.8: Segmentation results on challenging videos from FBMS without fine-tuning.

3.6.3 *Qualitative analysis on FBMS and ViSal*

In Figures 3.8 and 3.9, we visualize segmentation results on videos from the test sets of FBMS [27] and ViSal [28] respectively. The model is trained only with the DAVIS 2016 training set. We do not fine-tune it on the training set of FBMS or ViSal.



Figure 3.9: Segmentation results on challenging videos from ViSal without fine-tuning.

3.6.4 *Foreground correspondences analysis*

In Figure 3.10, we visualize more examples of foreground pixel correspondences to pixels in the anchor frame. Most pixels are randomly selected from the fore-



Figure 3.10: Similarity scores of a foreground pixel on a distant frame with pixels in the anchor frame. The left, middle, and right images from each video illustrate, respectively, the target frame with the sampled foreground pixel (marked by the blue cross), the anchor frame, and similarities overlaid on the anchor frame.

ground area on the last frame of the video (except when the foreground becomes too small in the last frame, in which case another frame is randomly chosen).

3.6.5 *Instance pruning*

Algorithm 1 details the instance pruning procedure. First, *SmallStatic* returns a set of bounding boxes and the corresponding instance masks that represent small and nearly static instances. Then, *GetPruningMask* takes these instances and the original masks as inputs, and generates a pruning mask per frame, which incorporates all small and static instances that are much smaller than the largest instance in the current frame. Finally, each input mask is multiplied element-wise with the corresponding pruning mask to output the final predictions.

3.7 STATEMENT OF AUTHORSHIP

The paper [89] from which this chapter originates is multi-authored, as is common with papers in science-based subjects. We specify the extent of contribution of each author as follows.

- Zhao Yang participated in the brainstorming of ideas, wrote code, ran experiments, and prepared the manuscript.
- Qiang Wang participated in the brainstorming of ideas, wrote code, and contributed to the preparation of the manuscript.
- Luca Bertinetto contributed to the preparation of the manuscript.
- Weiming Hu reviewed the manuscript.
- Song Bai participated in the brainstorming of ideas and reviewed the manuscript.
- Philip HS Torr organized the project, participated in the brainstorming of ideas, and reviewed the manuscript.

Algorithm 1 Instance Pruning

Input: original masks $X = [x_0, ..., x_{N-1}]$, bounding boxes/instance masks $E = [E_0, ..., E_{M-1}]$, for N frames and M total instances
Output: refined masks X'
$size_low \leftarrow Area(Sort(E)[-N])$
$T \leftarrow SmallStatic(E, 0.6, 0.5N, size_low)$
for $t = 1$ **to** N **do**
  Let b_t be instances on frame t from E
  $F \leftarrow GetPruningMask(x_t, T, size_low)$
  $x_t \leftarrow x_t \odot F$
end for
return X
function $SmallStatic(b, iou, support, size)$
  $sm_stat_instances \leftarrow \emptyset$
  for b_i in b **do**
    for b_j in b **do**
      $count \leftarrow 0$
      if $IoU(b_i, b_j) > iou$ **then**
        $count \leftarrow count + 1$
      end if
    end for
    if $count > support$ and $Size(b_i) < size$ **then**
      Add b_i to $sm_stat_instances$
    end if
  end for
  return $sm_stat_instances$
end function
function $GetPruningMask(x_t, T, s)$
  $pruning_mask \leftarrow \emptyset$, $target_size \leftarrow -\infty$
  $T_t \leftarrow Sort(T_t, descending)$
  if $Size(T_t[0]) > s$ & $Len(T_t) > 0$ & $Size(T_t[0]) > 2Size(T_t[1])$ **then**
    $target_size \leftarrow Size(T_t[0])$
  end if
  for all $T_t[i]$ in T_t **do**
    if $Size(T_t[i]) < \frac{target_size}{3}$ **then**
      $pruning_mask \leftarrow pruning_mask \cup T_t[i]$
    end if
  end for
  return $pruning_mask$
end function

4

LAVT: LANGUAGE-AWARE VISION TRANSFORMER FOR REFERRING IMAGE SEGMENTATION

4.1 ABSTRACT

Referring image segmentation is a fundamental vision-language task that aims to segment out an object referred to by a natural language expression from an image. One of the key challenges behind this task is leveraging the referring expression for highlighting relevant positions in the image. A paradigm for tackling this problem is to leverage a powerful vision-language ("cross-modal") decoder to fuse features independently extracted from a vision encoder and a language encoder. Recent methods have made remarkable advancements in this paradigm by exploiting Transformers as cross-modal decoders, concurrent to the Transformer's overwhelming success in many other vision-language tasks. Adopting a different approach in this work, we show that significantly better cross-modal alignments can be achieved through the early fusion of linguistic and visual features in intermediate layers of a vision Transformer encoder network. By conducting cross-modal feature fusion in the visual feature encoding stage, we can leverage the well-proven correlation modeling power of a Transformer encoder for excavating helpful multi-modal context. This way, accurate segmentation results are readily harvested with a lightweight mask predictor. Without bells and whistles, our method surpasses the previous state-of-the-art methods on RefCOCO, RefCOCO+, and G-Ref by large margins.

4.2 INTRODUCTION

Given an image and a text description of the target object, referring image segmentation aims at predicting a pixel-wise mask that delineates that object [57, 58]. It yields great value for various applications such as language-based human-robot interaction [193] and image editing [194]. In contrast to conventional single-modality visual segmentation tasks based on fixed category conditions [195, 196], referring image segmentation has to deal with the much richer vocabularies and syntactic varieties of human natural languages. In this task, the target object is inferred from a free-form expression, which includes words and phrases presenting the concepts of entities, actions, attributes, positions, *etc*., organized by syntactic rules. Therefore, the key challenge of this task is to exploit visual features that are relevant to the given text conditions.

There have been growing efforts devoted to referring image segmentation over the past few years. A widely adopted paradigm is to first independently extract vision and language features from different encoder networks, and then fuse them to make predictions with a cross-modal decoder. Concretely, the fusion strategies include recurrent interaction [61, 197], cross-modal attention [63, 65, 66], multi-modal graph reasoning [67], linguistic structure-guided context modeling [68], *etc*. Recent advances (*e.g*., [60]) bring performance improvements via employing a cross-modal Transformer [69] decoder (illustrated in Figure 4.1 (a)) to learn more effective cross-modal alignments, which is in concurrence with Transformer's overwhelming success in many other vision-language tasks [198–201].

Although great progress has been achieved, the potentiality of the Transformer for enhancing referring image segmentation is still far from being sufficiently explored in the conventional paradigm. Specifically, cross-modal interactions occur only after feature encoding, and a cross-modal decoder is solely responsible for aligning the visual and linguistic features. As a result, previous methods fail to effectively leverage the rich Transformer layers in the

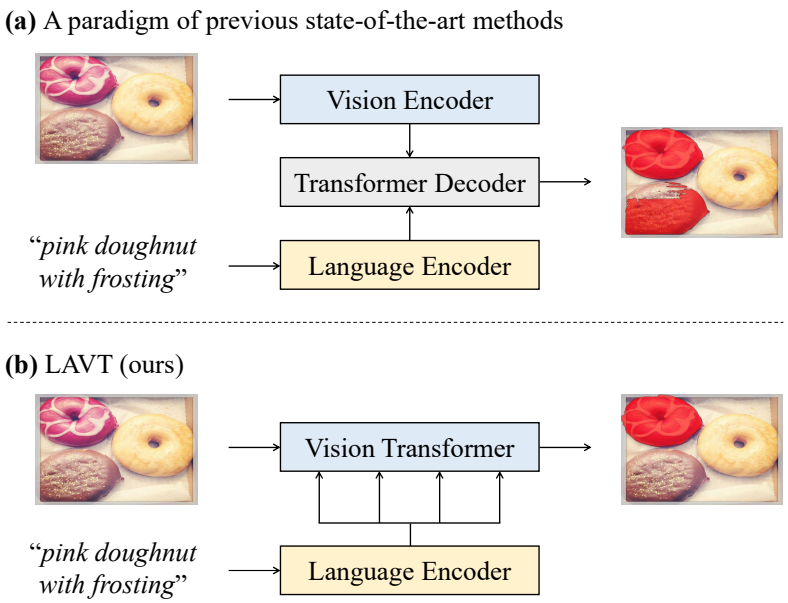


Figure 4.1: The task of referring image segmentation takes one image and one text description as inputs, and predicts a mask delineating the object specified in the description. (a) The previous state-of-the-art method (*i.e.*, VLT [60]) leverages a vision-language Transformer decoder for cross-modal feature fusion. (b) Conversely, we propose to directly integrate linguistic information into visual features at intermediate levels of a vision Transformer network, where beneficial vision-language cues are jointly exploited. A lightweight mask predictor can thus readily replace the complicated cross-modal decoder in previous counterparts.

encoder for excavating helpful multi-modal context. To address these issues, a potential solution is to exploit a visual encoder network for jointly embedding linguistic and visual features during visual encoding.

Accordingly, we propose a **L**anguage-**A**ware **V**ision **T**ransformer (**LAVT**) network, in which visual features are encoded together with linguistic features, being "aware" of their relevant linguistic context at each spatial location. As shown in Figure 4.1 (b), LAVT makes full use of the multi-stage design in a modern vision Transformer backbone network, leading to a hierarchical language-aware visual encoding scheme. Specifically, we densely integrate linguistic features into visual features via a pixel-word attention mechanism, which occurs at each stage of the network. The beneficial vision-language cues are then exploited by the following Transformer blocks, *e.g.*, [78], in the next encoder stage. This approach enables us to forgo a complicated cross-modal decoder, and instead harvest segmentation with a lightweight mask predictor.

To evaluate the effectiveness of the proposed method, we conduct extensive experiments on various mainstream referring image segmentation datasets. LAVT achieves 72.73%, 62.14%, 61.24%, and 60.50% overall IoU on the validation sets of RefCOCO [107], RefCOCO+ [107], G-Ref (UMD partition) [108], and G-Ref (Google partition) [109], improving the state of the art for these datasets by absolute margins of 7.08%, 6.64%, 6.84%, and 8.57%, respectively.

To summarize, our contributions are twofold:

- We propose LAVT, a Transformer-based referring image segmentation framework that performs language-aware visual encoding in place of cross-modal fusion post feature extraction.
- We achieve new state-of-the-art results on three datasets for referring image segmentation, demonstrating the effectiveness and generality of the proposed method. Source code is available at LAVT-RIS.

4.3 RELATED WORK

Referring image segmentation has attracted growing attention in the research community and there are two main processes in conventional pipelines: (1) extracting features from the text and image inputs respectively, and (2) fusing the multi-modal features to predict the segmentation mask. In the first process, previous methods adopt recurrent neural networks [49, 58, 61, 197, 202] and language Transformers [70, 83] to encode language inputs. To encode visual inputs, vanilla fully convolutional networks [13, 58, 61], DeeplabV3 [83, 104, 197], and DarkNet [113, 127, 202] have been successively employed in previous methods with the purpose of learning discriminative representations.

The multi-modal feature fusion module is the key component that prior arts focus on. For example, Hu *et al*. [58] propose the first baseline based on the concatenation operation, which is improved by Liu *et al*. [61] with a recurrent strategy. Shi *et al*. [63], Chen *et al*. [65], Ye *et al*. [64], and Hu *et al*. [66] model

cross-modal relations between language and vision features via various attention mechanisms. Yu *et al*. [129] and Huang *et al*. [67] leverage knowledge about sentence structures to capture different concepts (*e.g*., categories, attributes, relations, *etc*.) in multi-modal features, while Hui *et al*. [68] exploit syntactic structures among words for guiding multi-modal context aggregation.

The methods most related to ours are VLT [60] and EFN [128], where the former designs a Transformer decoder for fusing linguistic and visual features, and the latter adopts a convolutional vision backbone network for encoding language information. Different from [60], we propose an early fusion scheme which effectively exploits the Transformer encoder for modeling multi-modal context. Compared to [128], we do not rely on a complicated cross-modal decoder, which leads to a clearer and more effective framework. Under fair comparisons, our method outperforms these two counterparts by large margins.

Transformer is first introduced as a sequence-to-sequence deep attention-based language model[69], and has dominated the natural language processing (NLP) field [70, 72, 73] due to its strong capability on global context modeling. More recently, it has achieved great success on various computer vision tasks, *e.g*., image classification [74, 78, 203], action recognition [184, 204], object detection [75, 76, 78], and semantic segmentation [77, 78, 205].

There has also been a rich line of work on Transformers in the intersection area of computer vision and NLP [131, 206]. For example, Radford *et al*. devise a large-scale pretraining model, named CLIP [198], which applies contrastive learning [207–209] on features learned by a vision Transformer and a language Transformer. Hu *et al*. [199] propose a Unified Transformer (UniT) model that jointly learns multiple vision-language tasks across different domains. Besides, growing efforts have been devoted to other tasks such as visual question answering [200] and text-to-video retrieval [201]. However, to the best of our knowledge, there have been very few attempts on designing a unified Transformer model for the task of referring image segmentation.

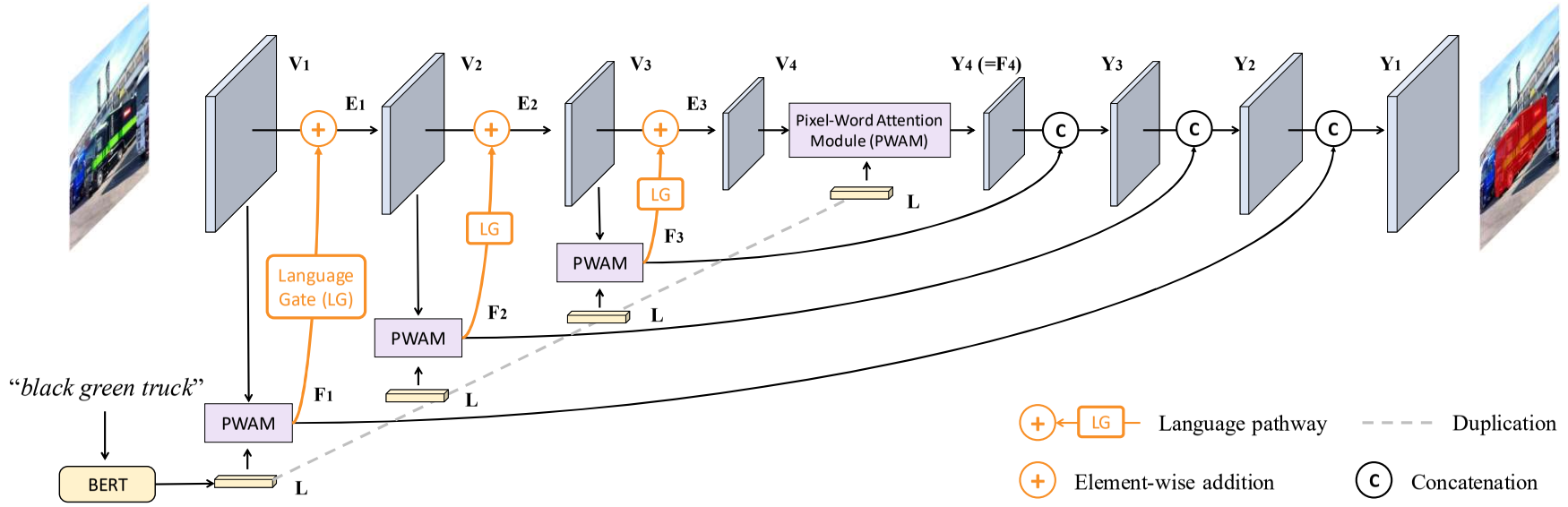


Figure 4.2: Overall pipeline of the proposed LAVT. We leverage a hierarchical vision Transformer [78] to perform language-aware visual encoding. At each stage, visual feature maps V_i, $i \in \{1,2,3,4\}$ are encoded from the corresponding stage of Transformer layers (which are described in Section 4.4.1 and for diagrammatic clarity, are not illustrated in this figure). Then V_i are used as queries for generating a set of position-specific language feature maps F_i, $i \in \{1,2,3,4\}$ in the pixel-word attention module (Section 4.4.2). Next, we adaptively fuse F_i with the original V_i via a language pathway (Section 4.4.3). The new visual feature maps E_i, $i \in \{1,2,3\}$ are then passed into the next stage of Transformer layers for further processing. A standard segmentation decoder head (Section 4.4.4) produces the final segmentation output.

4.4 METHOD

Figure 4.2 illustrates the pipeline of our Language-Aware Vision Transformer (LAVT), which leverages a hierarchical vision Transformer to jointly embed language and vision information to facilitate cross-modal alignments. In this section, we start by introducing our language-aware visual encoding strategy in Section 4.4.1, which is achieved with a pixel-word attention module detailed in Section 4.4.2 and a language pathway detailed in Section 4.4.3. Then in Section 4.4.4 we describe the light-weight mask predictor used to obtain final results.

4.4.1 *Language-aware visual encoding*

Given an input pair of an image and a natural language expression that specifies an object from the image, our model outputs a pixel-wise mask that delineates the object. To extract language features, we employ a deep language representation model to embed the input expression into high-dimensional word vectors. We denote the language features as $L \in \mathbb{R}^{C_t \times T}$, where C_t and T denote the number of channels and the number of words, respectively.

After obtaining the language features, we perform joint visual feature encoding and vision-language (which is also called "cross-modal" or "multi-modal" in the following contents) feature fusion through a hierarchy of vision Transformer layers organized into four stages. We index each stage using $i \in \{1, 2, 3, 4\}$ in the bottom-up direction. Each stage employs a stack of Transformer encoding layers (with the same output size) ϕ_i, a multi-modal feature fusion module θ_i, and a learnable gating unit ψ_i. Within each stage, language-aware visual features are generated and refined via three steps. First, the Transformer layers ϕ_i take the features from the previous stage as input, and output enriched visual features, denoted as $V_i \in \mathbb{R}^{C_i \times H_i \times W_i}$. Then, V_i are combined with language features L via the multi-modal feature fusion module θ_i to produce a set of multi-modal features, denoted as $F_i \in \mathbb{R}^{C_i \times H_i \times W_i}$. Finally,

each element in F_i is weighted by the learnable gating unit ψ_i and then added element-wise to V_i to produce a set of enhanced visual features embedded with linguistic information, which we denote as $E_i \in \mathbb{R}^{C_i \times H_i \times W_i}$. We refer to the computations in this final step as the language pathway. Here, C_i, H_i, and W_i denote the number of channels, the height, and the width of feature maps in the i-th stage, respectively.

The four stages of Transformer encoding layers correspond to the four stages in a Swin Transformer [78], which is an efficient hierarchical vision backbone suitable for addressing dense prediction tasks. The multi-modal feature fusion module within each stage is our proposed pixel-word attention module (PWAM), which is designed with the aim of densely aligning linguistic meanings with visual clues. And the gating unit is what we refer to as the language gate (LG), a special unit that we devise for regulating the flow of linguistic information along the language pathway (LP).

4.4.2 *Pixel-word attention module*

In order to separate a target object from its background, it is important to align the visual and linguistic representations of the object across modalities. One general approach is to combine the representation of each pixel with the representation of the referring expression, and learn multi-modal representations that are discriminative of a "referent" class and a "background" class. Previous approaches have developed various mechanisms for addressing this challenge, including dynamic convolutions [62], concatenations [58, 62, 197], cross-modal attentions [59, 63, 64, 66, 128], graph neural networks [148], *etc*. Compared to most of the previous cross-modal attention mechanisms [59, 63, 64, 66, 128], our pixel-word attention module (PWAM) produces a much smaller memory footprint as it avoids computing attention weights between two image-sized spatial feature maps, and is also simpler due to fewer attention steps.

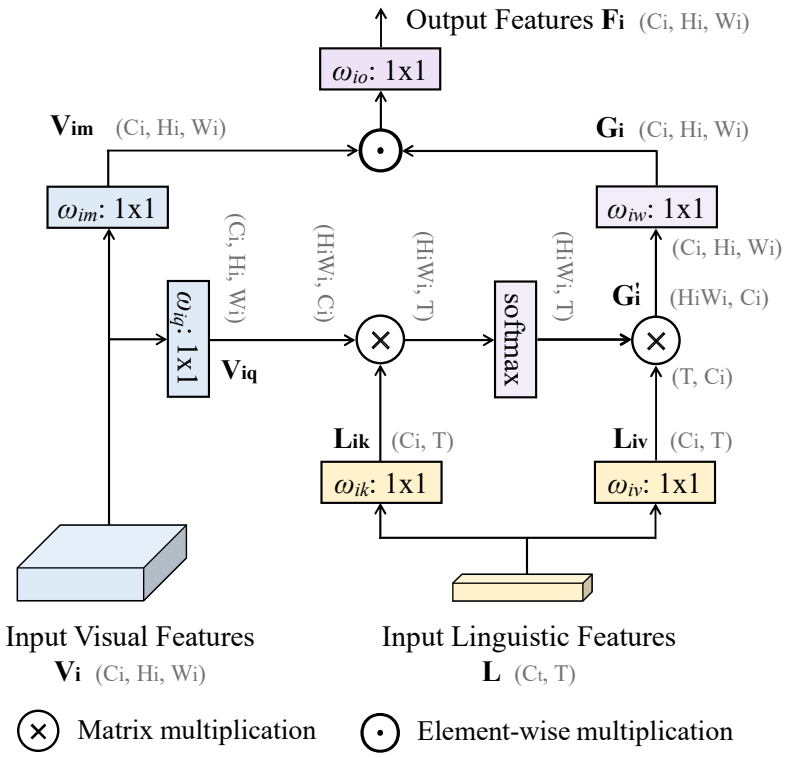


Figure 4.3: Pipeline of the pixel-word attention module (PWAM). First, a single-head scaled dot-product attention [69] is performed using the input visual feature maps V_i as queries and the input linguistic feature maps L as keys and values. The result, G_i, is a set of linguistic feature maps of the same spatial size as V_i. G_i is then multiplied element-wise with a projection of the input visual feature maps V_{im}, followed by another projection before final output. A detail which we found important empirically is the adoption of an instance normalization [210] layer in the projection functions ω_{iq} and ω_{iw} (see the text below and Table 4.3).

Figure 4.3 illustrates PWAM schematically. Given the input visual features $V_i \in \mathbb{R}^{C_i \times H_i \times W_i}$ and linguistic features $L \in \mathbb{R}^{C_t \times T}$, PWAM performs multi-modal fusion in two steps, as introduced in the following. First, at each spatial location, PWAM aggregates the linguistic features L across the word dimension to generate a position-specific, sentence-level feature vector, which collects linguistic information most relevant to the current local neighborhood. This step generates a set of spatial feature maps, $G_i \in \mathbb{R}^{C_i \times H_i \times W_i}$. Concretely, we

obtain G_i as follows

$$V_{iq} = \text{flatten}(\omega_{iq}(V_i)), \tag{4.1}$$

$$L_{ik} = \omega_{ik}(L), \tag{4.2}$$

$$L_{iv} = \omega_{iv}(L), \tag{4.3}$$

$$G_i' = \text{softmax}(\frac{V_{iq}^T L_{ik}}{\sqrt{C_i}}) L_{iv}^T, \tag{4.4}$$

$$G_i = \omega_{iw}(\text{unflatten}(G_i'^T)), \tag{4.5}$$

where ω_{iq}, ω_{ik}, ω_{iv}, and ω_{iw} are projection functions. Each of the language projections ω_{ik} and ω_{iv} is implemented as a 1×1 convolution with C_i number of output channels. And the query projection ω_{iq} and the final projection ω_{iw} each is implemented as a 1×1 convolution followed by instance normalization, with C_i number of output channels. Here, 'flatten' refers to the operation of unrolling the two spatial dimensions into one dimension in row-major, C-style order, and 'unflatten' refers to the opposite operation. These two operations and transposing are used to transform feature maps into proper shapes for calculation. Equations (4.1) and (4.5) implement the scaled dot-product attention [69] using visual features V_i as the query and linguistic features L as the key and the value, with instance normalization after linear transformation in the query projection function ω_{iq} and the output projection function ω_{iw}.

Second, after obtaining the linguistic features G_i which have the same shape as V_i, we combine them to produce a set of multi-modal feature maps F_i via element-wise multiplication. Specifically, our step is described as follows

$$V_{im} = \omega_{im}(V_i), \tag{4.6}$$

$$F_i = \omega_{io}(V_{im} \odot G_i), \tag{4.7}$$

where $\odot$ denotes element-wise multiplication and ω_{im} and ω_{io} are a visual projection and a final multi-modal projection, respectively. Each of the two functions is implemented as a 1×1 convolution followed by ReLU [211] nonlinearity.

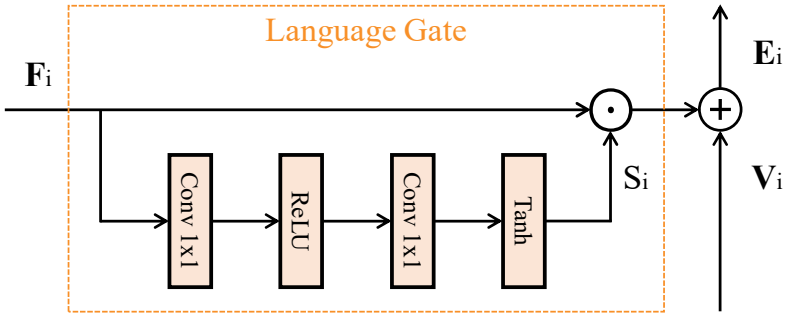


Figure 4.4: Schema of the language pathway, which leverages a language gate (LG) for regulating multi-modal information flow. LG is implemented as a two-layer perceptron.

4.4.3 *Language pathway*

As described earlier, at each stage, we merge the output from PWAM, F_i, with the output from the Transformer layers, V_i. We refer to the computations in this merging operation as the language pathway. In order to prevent F_i from overwhelming the visual signals in V_i and to allow an adaptive amount of linguistic information flowing to the next stage of Transformer layers, we design a language gate which learns a set of element-wise weight maps based on F_i to re-scale each element in F_i. The language pathway is schematically illustrated in Figure 4.4 and mathematically described as follows

$$S_i = \gamma_i(F_i), \tag{4.8}$$

$$E_i = S_i \odot F_i + V_i, \tag{4.9}$$

where $\odot$ indicates element-wise multiplication and γ_i is a two-layer perceptron, with the first layer being a 1×1 convolution followed by ReLU [211] nonlinearity and the second layer being a 1×1 convolution followed by a hyperbolic tangent function. As detailed in the ablation studies in Table 4.3, we have experimented with and without using a language gate along the language pathway, as well as different final nonlinear activation functions in the language gate, and found that using the gate with *tanh* final nonlinearity works the best for our model. The summation operation in Equation (4.9) is an effective way of utilizing pre-trained vision Transformer layers for multi-modal embedding, as the treatment

of multi-modal features as "supplements" (or "residuals") avoids disrupting the initialization weights pre-trained on pure vision data. We have observed much worse results in the case of adopting replacement or concatenation.

4.4.4 *Segmentation*

We combine the multi-modal feature maps, F_i, $i \in \{1, 2, 3, 4\}$, in a top-down manner to exploit multi-scale semantics for final segmentation. The decoding process can be described by the following recursive function

$$\begin{cases} Y_4 &= F_4, \\ Y_i &= \rho_i([v(Y_{i+1}); F_i]), \quad i = 3,\ 2,\ 1. \end{cases} \tag{4.10}$$

Here '[;]' denotes feature concatenation along the channel dimension, v represents upsampling via bilinear interpolation, and ρ_i is a projection function implemented as two 3×3 convolutions connected by batch normalization [212] and ReLU [211] nonlinearity. The final feature maps, Y_1, are projected into two class score maps via a 1×1 convolution.

4.4.5 *Implementation*

We implement our method in PyTorch [213] and use the BERT implementation from HuggingFace's Transformer library [214]. The Transformer layers in LAVT are initialized with classification weights pre-trained on ImageNet-22K [215] from the Swin Transformer [78]. Our language encoder is the base BERT model with 12 layers and hidden size 768 from [69] (hence C_t in Section 4.4 is 768) and is initialized using the official pre-trained weights. The rest of the weights in our model are randomly initialized. C_i in Section 4.4 is set to 512 and the model is optimized with cross-entropy loss. Following [78], we adopt the AdamW [216] optimizer with weight decay 0.01 and initial learning rate 0.00005 with polynomial learning rate decay. We train our model for 40 epochs with batch size 32. We iterate through each object (while randomly sampling one

referring expression for it) exactly once in an epoch. Images are resized to 480×480 and no data augmentation techniques are applied. During inference, *argmax* along the channel dimension of the score maps are used as predictions.

4.5 EXPERIMENTS

4.5.1 *Datasets and metrics*

We evaluate our method on three standard benchmark datasets, RefCOCO [107], RefCOCO+ [107], and G-Ref [108, 109]. Images in the three datasets are collected from the MS COCO dataset [195] and annotated with natural language expressions. Each of RefCOCO, RefCOCO+, and G-Ref contains 19,994, 19,992, and 26,711 images, with 50,000, 49,856, and 54,822 annotated objects and 142,209, 141,564, and 104,560 annotated expressions, respectively. Expressions in RefCOCO and RefCOCO+ are very succinct (containing 3.5 words on average). In contrast, expressions in G-Ref are more complex (containing 8.4 words on average), which makes the dataset particularly challenging. Conversely, RefCOCO and RefCOCO+ tend to have more objects of the same category per image (3.9 on average) compared to G-Ref (1.6 on average), therefore they better evaluate an algorithm's ability to comprehend instance-level details. A characteristic of RefCOCO+ is that location words are banned in its expressions, which also makes it more challenging. Additionally, there are two different partitions of the G-Ref dataset, one by UMD [108] and the other by Google [109]. We report results on both. When evaluating on each dataset, we train our model on the training set of that dataset. Finally, we make note of the ambiguities and foul language found in many expressions of RefCOCO with the hope that future community efforts will address them.

We adopt the common metrics of overall intersection-over-union (oIoU), mean intersection-over-union (mIoU), and precision at the 0.5, 0.7, and 0.9 threshold values. The overall IoU is measured as the ratio between the total

Method	Language Model	RefCOCO			RefCOCO+			G-Ref		
		val	test A	test B	val	test A	test B	val (U)	test (U)	val (G)
DMN [62]	SRU	49.78	54.83	45.13	38.88	44.22	32.29	-	-	36.76
RRN [197]	LSTM	55.33	57.26	53.93	39.75	42.15	36.11	-	-	36.45
MAttNet [129]	Bi-LSTM	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	-
CMSA [64]	None	58.32	60.61	55.09	43.76	47.60	37.89	-	-	39.98
CAC [217]	Bi-LSTM	58.90	61.77	53.81	-	-	-	46.37	46.95	44.32
STEP [65]	Bi-LSTM	60.04	63.46	57.97	48.19	52.33	40.41	-	-	46.40
BRINet [66]	LSTM	60.98	62.99	59.21	48.17	52.32	42.11	-	-	48.04
CMPC [67]	LSTM	61.36	64.53	59.64	49.56	53.44	43.23	-	-	49.05
LSCM [68]	LSTM	61.47	64.99	59.55	49.34	53.12	43.50	-	-	48.05
CMPC+ [148]	LSTM	62.47	65.08	60.82	50.25	54.04	43.47	-	-	49.89
MCN [127]	Bi-GRU	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
EFN [128]	Bi-GRU	62.76	65.69	59.67	51.50	55.24	43.01	-	-	51.93
BUSNet [218]	Self-Att	63.27	66.41	61.39	51.76	56.87	44.13	-	-	50.56
CGAN [59]	Bi-GRU	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	46.54
LTS [202]	Bi-GRU	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
VLT [60]	Bi-GRU	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65	49.76
LAVT (Ours)	BERT	**72.73**	**75.82**	**68.79**	**62.14**	**68.38**	**55.10**	**61.24**	**62.09**	**60.50**

Table 4.1: Comparison with state-of-the-art methods in terms of overall IoU on three benchmark datasets. U: The UMD partition. G: The Google partition. We refer to the language model of each reference method as the main learnable function that transforms word embeddings before multi-modal feature fusion. Interested readers can refer to the respective papers for embedding initialization and other details.

intersection area and the total union area of all test samples, each of which is a language expression and an image. This metric favors large objects. The mean IoU is the IoU between the prediction and ground truth averaged across all test samples. This metric treats large and small objects equally. The precision metric measures the percentage of test samples that pass an IoU threshold.

4.5.2 *Comparison with others*

In Table 4.1, we evaluate LAVT against the state-of-the-art referring image segmentation methods on the RefCOCO [107], RefCOCO+ [107], and G-Ref [108, 109] datasets using the oIoU metric. LAVT outperforms all previous methods on all evaluation subsets of all three datasets. Compared with the second-best method, VLT [60], LAVT achieves higher performance with absolute margins of 7.08%, 7.53%, and 6.06% on the validation, testA, and testB subsets of RefCOCO, respectively. Similarly, LAVT attains noticeable improvements over the previous state of the art on RefCOCO+ with wide margins of 6.64%, 9.18%, and 5.74% on

LP	PWAM	P@0.5	P@0.7	P@0.9	oIoU	mIoU
✓	✓	**84.46**	**75.28**	**34.30**	**72.73**	**74.46**
	✓	81.46	70.80	30.95	70.78	71.96
✓		81.76	72.76	32.46	71.03	72.31
		77.87	66.93	27.95	68.82	68.87

Table 4.2: Main ablation results on the RefCOCO validation set.

the validation, testA, and testB subsets, respectively. On the most challenging G-Ref dataset (which contains significantly longer expressions), LAVT surpasses the respective second-best methods on the validation and test subsets from the UMD partition by absolute margins of 6.84% and 5.44%, respectively. Similarly on the validation set from the Google partition, LAVT outperforms the second-best method EFN [128] by an absolute margin of 8.57%. This performance is achieved without using RefCOCO as additional training data in contrast to EFN.

4.5.3 *Ablation study*

We conduct several ablations to evaluate the effectiveness of the key components in our proposed network.

Language pathway (LP). Table 4.2 shows that removing LP (which corresponds to, mathematically, the removal of Equations (4.8) and (4.9), or schematically, the removal of the orange stream in Figure 4.2) leads to a drop of 1.95 and 2.50 absolute points in overall IoU and mean IoU, respectively. In addition, precision drops by 3 to 4 points across all three thresholds. These results demonstrate the benefit of exploiting our vision Transformer encoder network for jointly embedding linguistic and visual features.

Pixel-word attention module (PWAM). In this ablation study, we replace the spatial language feature maps G_i in PWAM with a sentence feature vector globally pooled from all words [91]. As shown in Table 4.2, this ablation leads to a drop of 1.70 and 2.15 absolute points in overall IoU and mean IoU, respectively, and a drop of 1 to 2 absolute points in precision across the three thresholds.

	P@0.5	P@0.7	P@0.9	oIoU	mIoU
(a) activation function in the language gate (LG)					
Tanh (*)	**84.46**	**75.28**	**34.30**	**72.73**	**74.46**
Sigmoid	81.89	72.71	33.35	70.49	72.47
(b) normalization layer in pixel-word attention module (PWAM)					
InstanceNorm (*)	**84.46**	**75.28**	**34.30**	**72.73**	**74.46**
LayerNorm	82.97	74.15	33.99	71.92	73.32
BatchNorm	82.89	73.82	33.53	71.59	73.09
None	81.91	72.73	33.11	70.66	72.34
(c) features used for final classification					
F_4, F_3, F_2, F_1 (G*)	**84.46**	**75.28**	34.30	**72.73**	**74.46**
F_4, F_3, F_2, F_1 (NG)	84.00	74.96	33.47	72.24	73.94
E_4, E_3, E_2, E_1 (G)	83.84	74.96	34.48	72.06	73.98
E_4, E_3, E_2, E_1 (NG)	84.33	74.94	**34.77**	72.27	74.12
V_4, V_3, V_2 (G)	83.36	74.47	32.61	71.38	73.29
V_4, V_3, V_2 (NG)	83.83	74.76	32.14	72.29	73.67
(d) multi-modal attention module					
PWAM (*)	**84.46**	**75.28**	**34.30**	**72.73**	**74.46**
BCAM [66]	82.26	72.81	33.31	70.19	72.42
GA (GARAN) [59, 127]	83.22	74.09	32.71	71.20	73.16

Table 4.3: Ablation studies on the RefCOCO validation set. (G) indicates that LG is adopted in the language pathway and (NG) indicates the opposite. Rows with (*) indicate default choices.

These results illustrate the effectiveness of densely aggregating linguistic context via our proposed attention mechanism for enhancing cross-modal alignments.

Activation function in the language gate (LG). Our proposed LG learns a set of spatial weight maps, which give our network the flexibility to control the flow of language information in the language pathway. In Table 4.3 (a), we compare the sigmoid function and the hyperbolic tangent function as the final activation function in LG. Using the sigmoid function leads to inferior results.

Normalization layer in PWAM. As described in Section 4.4.2, we adopt a final instance normalization layer in the projection functions ω_{iq} and ω_{iw} in PWAM. As we illustrate in Table 4.3 (b), this particular choice of normalization function has a non-trivial effect. In addition to instance normalization (our default choice), we experiment with batch normalization, layer normalization, and without having a normalization layer in the functions ω_{iq} and ω_{iw}. All three other choices lead to 1 to 2 absolute points drop in the overall IoU and mean

IoU metrics. Among these three choices, using batch normalization or layer normalization produces better results than not using a normalization layer.

Features used for prediction. As shown in Figure 4.4, the language-aware visual encoding process of LAVT produces three kinds of spatial feature maps which encapsulate visual and linguistic information, *i.e.*, the outputs from PWAMs (F_i, $i \in \{1,2,3,4\}$), the outputs from the Transformer layers (V_i, $i \in \{2,3,4\}$), and the inputs to the following Transformer layers (E_i, $i \in \{1,2,3\}$). While our default choice is to use F_i for predicting the object mask, we also consider the other two types of feature maps natural candidates for this purpose. As shown in Figure 4.2, E_4 is not generated in the standard architecture of LAVT. To have a convincing ablation study, we compute E_4 with an additional language pathway as defined in Equations (4.8) and (4.9). Therefore, we use E_i, $i \in \{1,2,3,4\}$ to predict the segmentation masks. In comparison, as multi-modal information has been progressively integrated into V_2, V_3, and V_4 along the bottom-up computation pathway while V_1 contains pure visual information, we do not use V_1 for prediction. In Table 4.3 (c), we report segmentation results when using each type of features with and without our proposed LG (indicated by "G" and "NG", respectively). Table 4.3 (c) shows that using our default choice of F_i with LG produces the best overall results among all choices. Also, we observe that while LG has a positive effect when using F_i for segmentation, it slightly degrades the results when E_i (72.06% vs. 72.27% in oIoU) or V_i (71.38% vs. 72.29% in oIoU) are used for segmentation.

Multi-modal attention module. In Table 4.3 (d), we compare PWAM with two state-of-the-art attention modules by directly replacing PWAM with them in our framework, using the same backbone, language model, and training recipes. Compared to both the grouped attention (GA or GARAN) [59, 127] and the bi-directional cross-modal attention module (BCAM) [66], PWAM achieves higher scores across all metrics. Note that BCAM is representative of the computationally-heavy attention modules and GA is the most recent top-performing module.

Method	P@0.5	P@0.7	P@0.9	oIoU	mIoU
LTS (Swin-B+BERT) [202]	80.59	69.48	26.13	69.94	70.56
EFN (Swin-B+BERT) [128]	82.55	73.27	31.68	70.76	72.95
VLT (Swin-B+BERT) [60]	83.24	72.81	24.64	70.89	71.98
Ours + VLT [60]	**84.57**	75.14	26.36	72.12	73.57
Ours	84.46	**75.28**	**34.30**	**72.73**	**74.46**

Table 4.4: Comparison between our method, LTS [202], VLT [60], and EFN [128] on the RefCOCO validation set, where all models use the same backbone, language model, and training recipes.

Fair comparison with reference methods. To further validate the effectiveness of our proposed method of fusing cross-modal information via a vision Transformer encoder network, in Table 4.4, we provide fair comparisons between our method and three previous state-of-the-art methods, LTS [202], VLT [60] and EFN [128]. All models use $\text{BERT}_{\text{BASE}}$ as the language encoder and Swin-B as the vision backbone network, following the same training settings (described in Section 4.4.5). While LTS employs a "locate-then-segment" pipeline, VLT is representative of methods that employ a cross-modal Transformer decoder. Conversely, EFN is representative of methods which fuse cross-modal information via an encoder network and additionally rely on a complicated decoder for obtaining the best results. As shown in Table 4.4, our method outperforms LTS, VLT, and EFN on the validation set of RefCOCO across all metrics. To further verify that our proposed LAVT encoding scheme is more effective than its counterpart cross-modal decoder approach, we combine our approach with VLT by substituting our original light-weight mask predictor with the cross-modal Transformer decoder from VLT. As shown in this experiment (indicated by "ours + VLT" in Table 4.4), employing a Transformer decoder to perform additional cross-modal feature fusion after language-aware visual encoding by LAVT generally does not bring extra gains (except a marginal 0.11% improvement in P@0.5).

Visualized predictions. In Figure 4.5, we visualize the predictions and feature maps of our full model and two ablated models (without the language pathway ("w/o LP") and without the pixel-word attention module ("w/o PWAM"), respectively). From the first row, we can observe that the higher-level feature

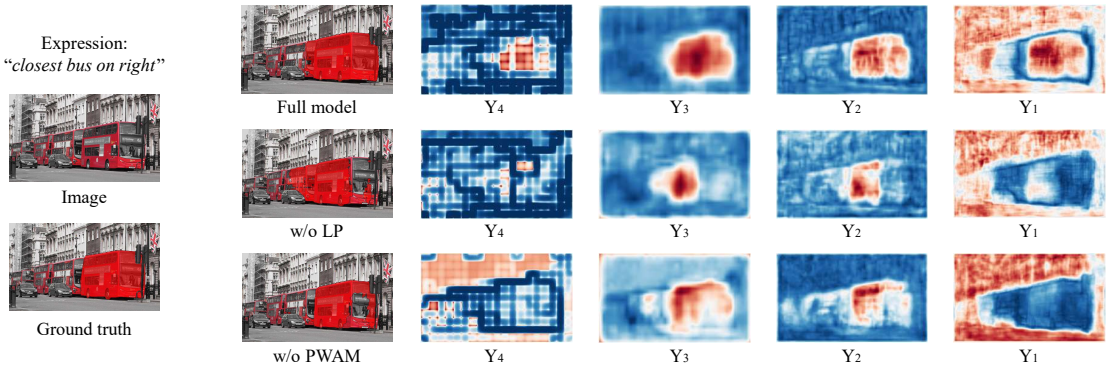


Figure 4.5: Visualized predictions and feature maps on an example from the RefCOCO validation set. From top to bottom, the left-most column illustrates the input expression, the input image, and the ground-truth mask overlaid on the input image. In each row, we visualize the predicted mask and the feature maps used for final classification (*i.e.*, Y_4, Y_3, Y_2, and Y_1) from left to right. LP represents the language pathway and PWAM represents the pixel-word attention module.

maps (*i.e.*, Y_4, Y_3, Y_2) in our full model can accurately locate the semantic concept given in text, while the low-level feature maps (*i.e.*, Y_1) contain rich boundary information important to binary segmentation. Comparing the predicted masks between the three models, we can observe that the removal of LP and the removal of PWAM both lead to false negative predictions on the front window area of the target bus, while the removal of LP additionally results in the false positive identification of the middle bus. These qualitative results further validate the effectiveness of our proposed LP and PWAM mechanisms. More example visualizations are shown in Figure 4.6.

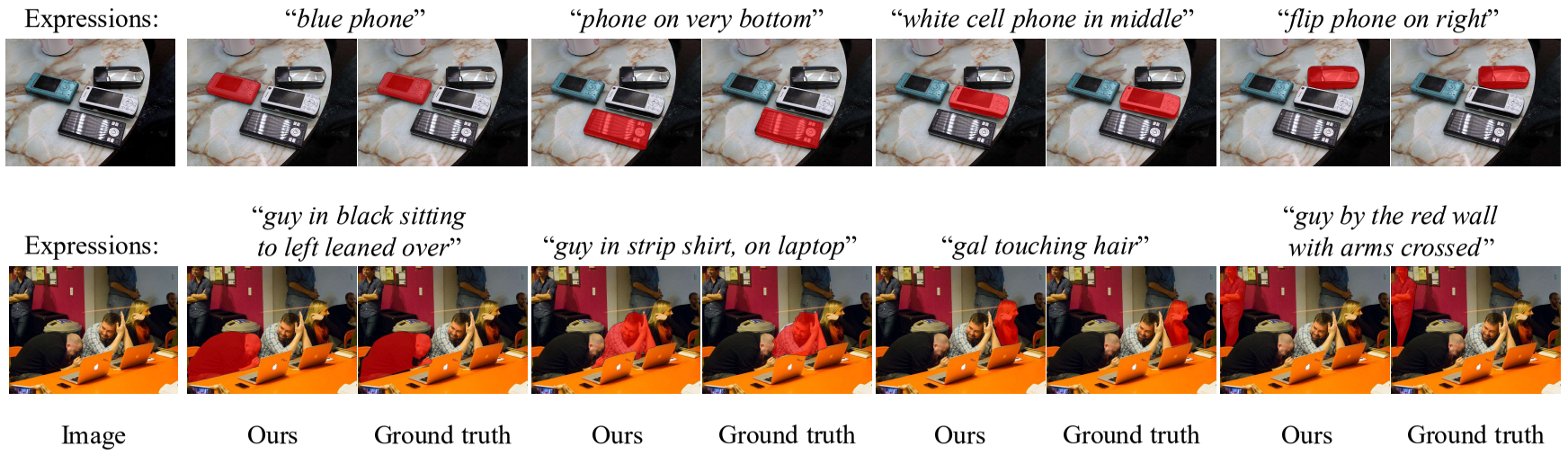


Figure 4.6: Visualizations of our predicted masks and the ground-truth masks on two examples from the RefCOCO validation set.

4.6 DISCUSSION

In this paper, we have proposed the Language-Aware Vision Transformer (LAVT), a framework for addressing the problem of referring image segmentation. Unlike previous methods, LAVT leverages the intermediate layers of the visual Transformer encoder network to jointly embed linguistic and visual features, thereby relocating the key process of cross-modal feature fusion to the stage of image/video encoding. This is achieved by "injecting" linguistic cues into the vision Transformer network at every stage, where the cues are aligned with visual features by a pixel-word attention module (PWAM) and modulated by a gating function. Extensive ablation studies validate our design choices, and experiment results on three challenging benchmarks demonstrate the advantage of our method with respect to the state-of-the-art approaches.

Since LAVT is designed to be a simple baseline for the referring image segmentation task, there can potentially be many interesting extensions from the current model. We briefly outline two in the following. First, beyond the currently proposed language-aware visual encoding scheme based on a vision Transformer, the encoding scheme that works in the opposite direction—vision-aware language encoding based on a language Transformer—may also produce strong features conducive to segmentation. If features from bi-directional fusion can be combined and complement each other, then this approach may potentially lead to better results. Second, from an architectural point of view, currently the PWAM at each stage amounts to a cross-attention layer (followed by an element-wise multiplication operation) inserted at the end of the corresponding stage of the Swin Transformer [78], effectively converting the original vision Transformer into a multi-modal Transformer. This leads to the question that whether we can scale up our method by making native changes to the Swin Transformer, where cross-modal attention and element-wise multiplication can be performed by exploiting the original non-overlapping window and shifted window attention layers in the Swin Transformer. This fully integrated approach,

if successful, has the potential to serve as a much more general architecture suitable for many vision-language tasks, including pre-training tasks.

Looking beyond the architectural perspective, there can potentially be many other improvements made to the existing state-of-the-art referring image segmentation models from different directions. As much of the interest in referring image segmentation stems from a desire to develop an algorithm that can segment an image or video in an open world (*i.e.*, segment out *any* object), it makes sense to explore the use of pre-trained large vision-language models (*e.g.*, CLIP [198] and Stable Diffusion [219]) to improve generalization. Some direct benefits of doing so may be inferred from the following two aspects. First, the enlarged vocabulary of the language encoder may vastly expand the number of concepts understood by the model. Second, alignment information in the pre-trained model should be useful. For instance, the pre-trained weights can be used for initialization, or weak supervisory signals may be extracted for the supervision of the task-specific model. In addition, we think that the successful exploitation of pre-trained representations depends crucially on our understanding of their nature, which differs for discriminative ones (*e.g.*, from CLIP) and generative ones (*e.g.*, from Stable Diffusion), and that like in many fields of artificial intelligence, acquiring this understanding may require a sustained research effort of an empirical nature. In Section 6.1, we discuss more about these thoughts.

Another interesting aspect of potential improvement based on empirical observations is the model's counting ability, especially in cluttered scenes. It is not uncommon for language expressions to use positional information to identify the target, for instance, "the fifth boat on the dock from the left" or "the third bottle from the right." Given the experience of working with deep neural networks for years, we think that a rule-based approach, or one implementing specialized, heuristics-based components in a network, won't be the best approach to this problem as they would not generalize well. Instead, a data-driven approach is more promising, *i.e.*, to enforce the network to learn

to count by using training data that require counting. Specifically, we could collect images that contain many objects of the same class, and annotate each and every object in an image with referring expressions that entail counting. The only variation among the expressions would be the position of the object, which would also be the only information useful for telling the objects apart. This way, the model may be trained to learn to count. The basis of this recommendation is the rule of thumb that a neural net is generally able to solve an association problem if sufficient data and correct supervision are given.

4.7 APPENDICES

4.7.1 *Potential biases of the language model*

There have been studies that suggest there exist ethnic biases in the pre-trained language model, BERT [70]. We refer interested readers to the work of Ahn *et al*. [220] for details, in which different kinds (including racial, gender, geological, *etc*.) of ethnic biases are analyzed and mitigation methods are proposed.

4.7.2 *The language pathway*

The motivation for our language pathway was to find a way to allow the vision Transformer layers to embed multi-modal information effectively. As a result, we built the language pathway as a residual connection [141, 221], which has been shown effective for combining features containing different types of information in a deep neural network. And the design of the language gate is inspired by previous work that featured learnable gates for regulating information flow in deep neural networks, such as the LSTM [49], the SENet [222], and CFBI [223].

Method	P@0.5	P@0.7	P@0.9	oIoU	mIoU
*Replacement (w/o LG)	–	–	–	–	–
*Concatenation (w/o LG)	72.89	58.15	20.02	60.52	63.41
Sum (w/o LG)	84.00	74.96	33.47	72.24	73.94
Sum (with LG; default)	**84.46**	**75.28**	**34.30**	**72.73**	**74.46**

Table 4.5: Design alternatives for the language pathway (annotated with the asterisk). 'LG' is short for language gate. '–' indicates extremely slow convergence during training.

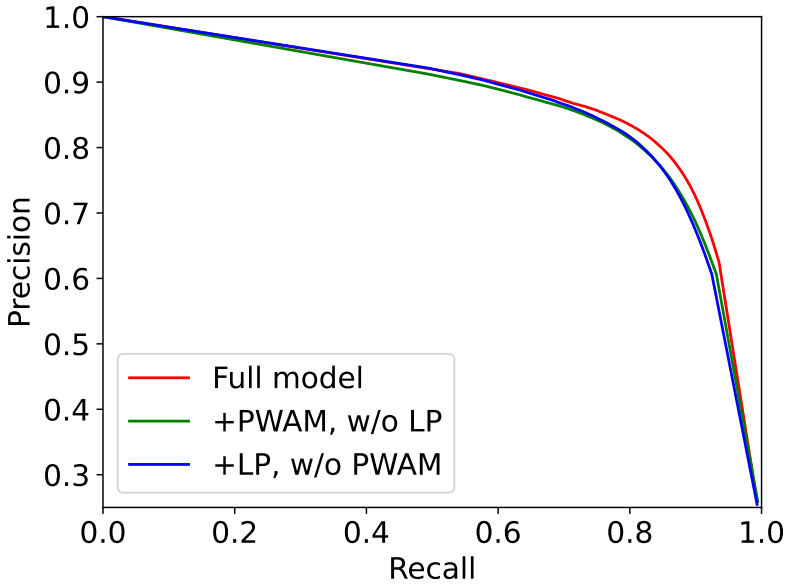


Figure 4.7: Precision-recall (PR) curves on the RefCOCO validation set. The full model obtains the best PR trade-off compared to the ablated models. When closely observing the "+LP" model (blue) and the "+PWAM" model (green), we can find that LP maintains a slight advantage in precision over PWAM up until around 0.8 recall.

4.7.3 *Precision-recall analysis*

To understand the precision-recall trade-off of LAVT and two of its ablated models, in Figure 4.7 we compute and plot the average precision and the average recall of all test samples in the validation set of RefCOCO at 100 thresholds evenly spaced out from 0 to 1 (where the prediction for a pixel is positive if the softmax-normalized score map of the object class exceeds the threshold and is negative otherwise).

4.7.4 *Mean IoU*

Method	Language Model	RefCOCO			RefCOCO+			G-Ref		
		val	test A	test B	val	test A	test B	val (U)	test (U)	val (G)
LAVT (Ours)	BERT	74.46	76.89	70.94	65.81	70.97	59.23	63.34	63.62	63.66

Table 4.6: Mean IoU of LAVT on the three benchmark datasets. These results complement the overall IoU reported in Table 4.1. Since mean IoU treats each object equally and does not favor large objects (as overall IoU does), we consider it a fairer metric and recommend more of its use for evaluating this task in the future.

4.7.5 *Visualizations*

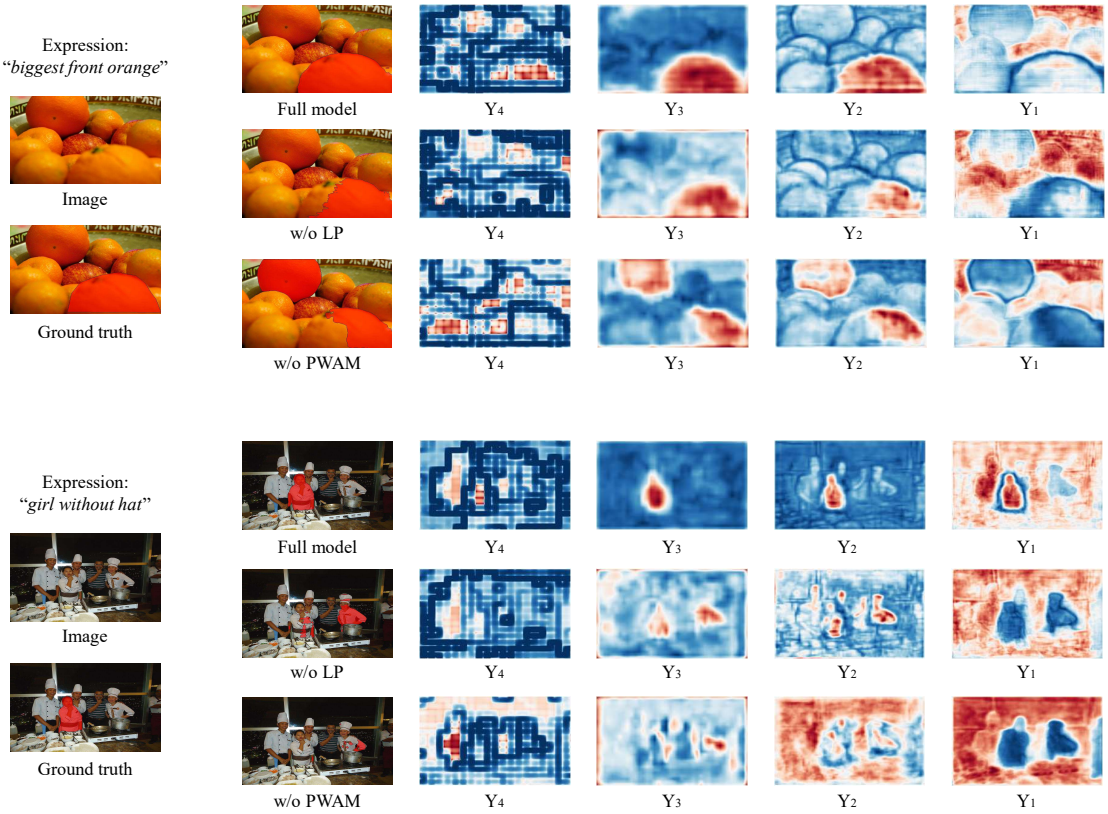


Figure 4.8: Additional visualizations of predictions and feature maps from the RefCOCO validation set. For each example, the left-most column illustrates the input expression, the input image, and the ground-truth mask overlaid on the input image. In each row, we visualize the predicted mask and the feature maps used for final classification (*i.e.*, Y_4, Y_3, Y_2, and Y_1) from left to right. LP represents the language pathway and PWAM represents the pixel-word attention module.



Figure 4.9: Visualizations of our predicted masks and the ground-truth masks on examples from the RefCOCO validation set. Examples enclosed with green lines are successful cases, and those enclosed with red lines are failed cases. In the successful cases, our predictions are nearly identical to the ground truth and are sometimes more accurate than the ground truth (see the second example from the right column, where part of the body of the man behind the chair is missing in the annotation). Among the two demonstrated failure cases, the first one is caused by ambiguity in the given expression (there are two boys that are on a skateboard and our model segments out both) and the second one is caused by our model's lack of knowledge of what a "pac man" is (obviously having not played the game Pac-Man, our model fails to associate the shape of the pizza to the shape of a Pac-Man).

4.8 STATEMENT OF AUTHORSHIP

The paper [90] from which this chapter originates is multi-authored, as is common with papers in science-based subjects. We specify the extent of contribution of each author as follows.

- Zhao Yang participated in the brainstorming of ideas, wrote code, ran experiments, and prepared the manuscript.
- Jiaqi Wang participated in the brainstorming of ideas and contributed to the preparation of the manuscript.
- Yansong Tang participated in the brainstorming of ideas and contributed to the preparation of the manuscript.
- Kai Chen reviewed the manuscript.
- Hengshuang Zhao participated in the brainstorming of ideas and reviewed the manuscript.
- Philip HS Torr organized the project, participated in the brainstorming of ideas, and reviewed the manuscript.

5

HIERARCHICAL INTERACTION NETWORK FOR VIDEO OBJECT SEGMENTATION FROM REFERRING EXPRESSIONS

ABSTRACT

In this chapter, we investigate the problem of video object segmentation from referring expressions (VOSRE). Conventional methods typically perform multi-modal fusion based on linguistic features and visual features extracted from the top layer of the visual encoder, which limits these models' ability to represent multi-modal inputs at different semantic and spatial granularity levels. To address this issue, we present an end-to-end hierarchical interaction network (HINet) for the VOSRE problem. Our model leverages the feature pyramid produced by the visual encoder to generate multiple levels of multi-modal features. This allows a more flexible representation of various linguistic concepts (*e.g.*, object attributes and categories) in different levels of the multi-modal features. Moreover, we further extract signals of moving objects from optical flow input, and utilize them as complementary cues for highlighting the referent and suppressing the background with a motion gating mechanism. In contrast to previous methods, this strategy allows our model to make online predictions without requiring the whole video as input. Despite its simplicity, our proposed HINet improves the state of the art for the DAVIS-16, DAVIS-17, and J-HMDB datasets, demonstrating its effectiveness and generality.

5.1 INTRODUCTION

As a fundamental task in computer vision, video object segmentation (VOS) has various real-world applications including augmented reality [224], robotics [36], video surveillance [37], and video editing [38]. Depending on the level of supervision provided at test time, the VOS problem has traditionally been tackled in the *semi-supervised* (initialized by pixel-level masks or scribbles) [16, 31] or *unsupervised* settings [16, 32]. In comparison, the recently introduced task of video object segmentation from referring expressions (VOSRE) aims to segment an object from a video with the guidance of a natural language description. This setting offers several advantages: (1) Language annotations are much more economical and scalable than pixel-level masks [84, 140], while they are also more intuitive (compared to scribbles) for providing target identification, especially on mobile devices. (2) Different from the unsupervised setting where no target specification is given, language queries greatly reduce the ambiguity in problem definition.

Due to the reasons above, VOSRE has attracted growing attention in recent years. With the purpose of aligning semantic concepts between language and vision, most of the existing methods focus on developing new mechanisms for combining linguistic and visual features, (*e.g.*, dynamic convolutions [79–81], cross-modal attentions [82], concatenation [58, 82, 83], summation [84], and multiplication [83]). However, they only extract visual feature maps from a single layer of the visual encoder. Despite having the potential of effectively fusing language and vision information, they have not exploited the fact that convolutional networks (*e.g.*, VGG [225], ResNet [141] and I3D [86]) encode visual features in a hierarchical way, where the scale, complexity, and abstractness of encoded features grow progressively from the bottom level to the top level [226, 227].

Figure 5.1: The goal of VOSRE is to predict a pixel-level mask, in every frame, that delineates the object described by a language query. Masks in this example are our predictions.

In related subfields, such as object detection and segmentation, the importance of utilizing not only top-level feature maps but also an array of feature maps from different levels is well known and has been widely explored. For instance, Feature Pyramid Network [221] extracts region-of-interest features from the outputs of five different stages of the encoder in order to capture objects of different scales, while PSPNet [228] and DeepLabV3+ [18] combine low-level and intermediate features with high-level features for recovering more accurate spatial and appearance details. As for our problem, a phrase typically describes multiple concepts about the referent object, such as its category, attributes, action, *etc*. These different concepts may not all be best represented by the same level of visual feature maps, *e.g*., clues for a described color may be more evident in lower-level features, and those for a category are more likely to be embedded in high-level features.

Motivated by these observations, we propose a hierarchical interaction network (HINet) to learn more effective multi-modal representations for VOSRE. The HINet incorporates a separate set of linguistic features with the visual feature maps from the different convolutional blocks of the visual encoder. This simple strategy generates multi-level, multi-modal representations at different semantic and spatial granularity levels, which greatly improves the accuracy of object localization. Furthermore, unlike previous methods which typically employ 3D CNNs [79, 81, 82] or space-time attention networks [84] to extract temporal information from multiple frames, we extract motion signals of moving objects from optical flow input and directly exploit them as complementary source of guidance for object localization. Specifically, we utilize optical flow magnitude in two ways: (1) we concatenate it to the RGB input to *implicitly* embed motion information in the resulting visual feature pyramid; (2) we encode it into high-dimensional features and use them as gating values to *explicitly* strengthen multi-modal features of moving objects and suppress those of static regions. We conduct experiments on the DAVIS-16 [16], DAVIS-17 [30], A2D [139], and J-HMDB [229] datasets. Despite its simplicity and

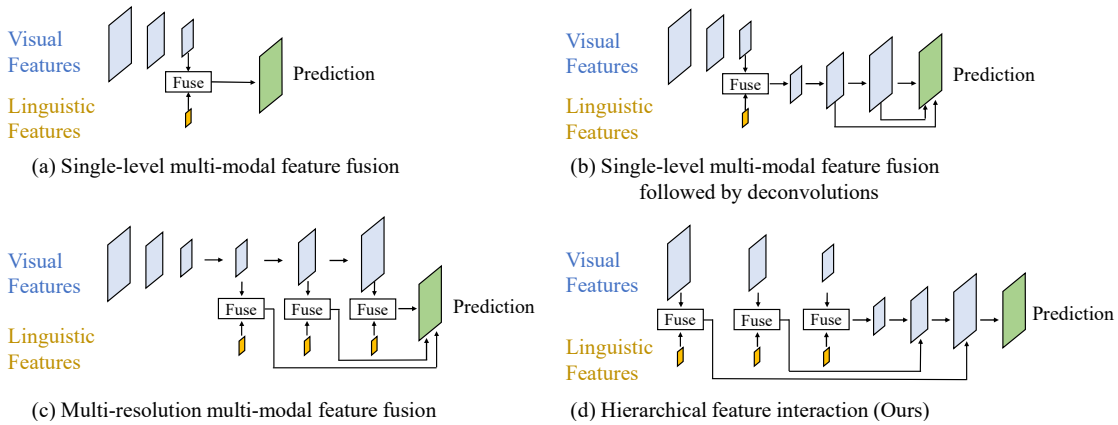


Figure 5.2: Comparison of different multi-modal feature fusion methods for VOSRE. (a) Single-level multi-modal feature fusion [58, 80, 81, 83]. (b) Single-level multi-modal feature fusion followed by deconvolutions [84]. (c) Multi-resolution multi-modal feature fusion in which **only the top-level visual features** are upsampled to multiple resolutions and then combined with linguistic features [79, 82]. (d) Our hierarchical feature interaction approach, which combines linguistic features with visual features from **multiple levels of the encoder.**

speed, our method improves the state of the art on three datasets and performs competitively on the other.

5.2 RELATED WORK

Referring image segmentation. In the image domain, a variety of methods for referring object segmentation perform multi-modal feature fusion by different strategies, *e.g.*, concatenation [58], recurrent interaction [61, 197], cross-modal attention [63, 65, 66, 143], multi-modal graph reasoning [67], and linguistic structure-guided context modeling [230]. Different from previous methods which leverage multi-level features for multi-modal fusion[65–67, 129, 143, 230], we further integrate motion information from optical flow at different levels of the network to tackle the more challenging VOSRE task in the video domain.

VOS. The VOS problem is traditionally tackled in the semi-supervised or the unsupervised setting. For semi-supervised VOS, the "supervision" refers to human guidance at test time and comes in the form of pixel-level masks or scribbles in the first frame [30]. While early methods [56, 92, 95–97, 101]

apply an online fine-tuning strategy to adapt to object appearance at test time, they are particularly slow and impractical in many use cases. Instead, recent methods directly propagate the first-frame annotations to the rest frames via sequential modeling approaches, *e.g.*, the space-time memory network [142], collaborative foreground-background integration [231], and many others [54, 55, 232]. For unsupervised VOS, there is no human supervision available at test time. Early benchmarks [16, 27, 28] assume a single target object per video and the methods rely on optical flow [40, 45, 47], LSTM [39], or cross-frame pixel similarities [89] for modeling appearance changes of the target object. Recent benchmarks [32] address the more practical and challenging multiple-object scenario. Researchers generally first predict an initial set of masks by an off-the-shelf instance segmentation model (*e.g.*, Mask R-CNN [14]), and then associate individual mask predictions across different frames into consistent object tracks [233–235].

VOSRE. Figure 5.2 displays a comparison of recent representative VOSRE approaches and our method in the multi-modal fusion stage. Note that we group the language-guided actor-action segmentation (L-AAS) methods [79–82, 148, 150, 153] into the category of VOSRE, since the only minor difference is that expressions in L-AAS put more emphasis on describing what an actor is doing in each video. Besides the multi-modal fusion strategy, we further design a gating mechanism to strengthen the moving object compared with [140], which also utilizes optical flow as input. Moreover, our HINet does not require any iterative refinement procedure at inference time compared with [84].

5.3 METHOD

Figure 5.3 illustrates the pipeline of our proposed *hierarchical interaction network* (HINet). Specifically, it employs three dedicated branches to extract high-dimensional features from the language query input, the video frame input, and the optical flow magnitude input, respectively. To effectively align language

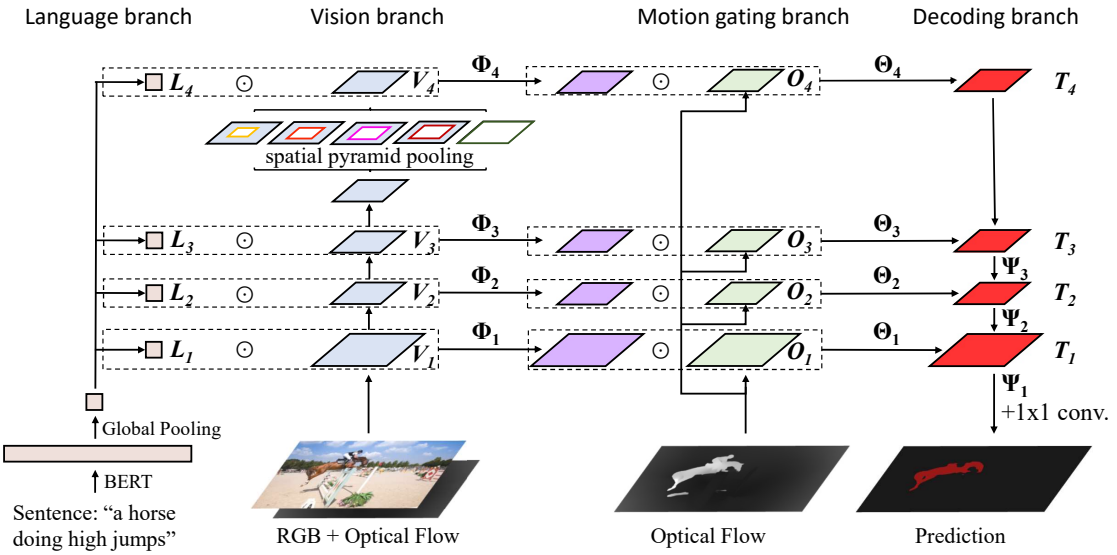


Figure 5.3: Schematic illustration of our approach. We combine the multi-modal features with a hierarchical feature interaction strategy, which allows flexible integration of various concepts between the language and vision data, and optical flow information further helps for highlighting moving objects. In this figure, $\odot$ denotes element-wise multiplication. Φ_i and Θ_i are 1×1 convolutions, where $i = 1, 2, 3, 4$. Ψ_j is two 3×3 convolutions connected by batch normalization and ReLU nonlinearity, where $j = 1, 2, 3$.

and vision semantics at different scales, we first fuse the extracted visual and linguistic features at multiple levels to obtain the "language-vision" feature maps. Then we extract multiple scales of motion feature maps from our motion gating branch and fuse them with "language-vision" feature maps to produce our final sets of multi-level, multi-modal feature maps, which capture motion information about objects that are moving. Finally, the multi-level, multi-modal feature maps are fused in a top-down manner for mask prediction. In the following, we describe our pipeline in more detail.

5.3.1 *Multi-modal feature encoding*

Encoding visual features. The input to our vision branch is a 4-channel image, which is a concatenation of the 3-channel RGB frame with the 1-channel optical flow magnitude image. We compute the optical flow magnitude after subtracting the mean flow vector from the optical flow field. We use the ResNet-

101 [141] network and an Atrous Spatial Pyramid Pooling (ASPP) module [104] to extract multiple levels of visual feature maps and apply a 1×1 convolution for channel reduction to each level of features. The multi-level visual features, $V_i \in \mathbb{R}^{c_i \times h_i \times w_i}$, $i \in \{1,2,3,4\}$, correspond to outputs from the first three residual blocks in ResNet and the output from the ASPP module respectively, where c_i, h_i, and w_i refer to the number of channels, the height, and the width of the corresponding feature maps.

Encoding language features. We employ BERT [70], a deep language representation model pre-trained from unlabeled text to extract a linguistic feature matrix, $G \in \mathbb{R}^{l \times c_s}$, from the input sentence. Here, l denotes the number of words and c_s denotes the number of channels. To generate sentence-level feature vectors for each level of the visual feature maps, we apply average pooling to G along the word dimension, and then apply four independent linear layers followed by batch normalization operations. We denote the output sentence-level feature vectors as $L_i \in \mathbb{R}^{c_i}$, $i \in \{1,2,3,4\}$, where c_i denotes the number of channels and are the same ones defined for the visual feature maps.

Encoding optical flow features. We employ four independent 7×7 convolutional layers with stride 1 to extract four sets of motion feature maps from the optical flow magnitude image input (same as the one described earlier). Each of the four sets of motion feature maps are downsampled to the same spatial resolution as the corresponding visual feature maps. We denote the output motion feature maps as $O_i \in \mathbb{R}^{c_i \times h_i \times w_i}$, $i \in \{1,2,3,4\}$, where c_i, h_i, and w_i denote the number of channels, the height, and the width of the feature maps. These notations are consistent with those defined for the linguistic and visual features.

5.3.2 *Hierarchical feature interaction*

To effectively utilize various linguistic concepts (*e.g.*, object attributes and categories) for the VOSRE task, we fuse the multi-modal representations at different semantic and spatial granularity levels with a hierarchical feature

interaction strategy. To achieve this, we first combine the extracted vision and language features, V_i and L_i, to generate "language-vision" multi-modal features. Then we fuse them with the motion features, O_i, to generate the final multi-modal feature maps. Concretely, our step to combine the vision and language features is described mathematically as follows

$$F_i' = \Phi_i(V_i \odot L_i), \quad i = 1,\ 2,\ 3,\ 4. \tag{5.1}$$

Here $\odot$ denotes element-wise multiplication, and Φ_i is a 1×1 convolution applied for feature fusion with the same number of input and output channels. The language feature vector L_i is element-wise multiplied with the visual feature vector at each spatial location on the visual feature maps V_i. Note that as described in the previous section, V_i and L_i have the same number of channels, which makes them compatible for element-wise multiplication. The 1×1 convolution Φ_i is employed to project the results to a different high-dimensional space. On the other hand, we combine the obtained "language-vision" features with the motion features by utilizing a motion gating branch, which aims to highlight the moving objects and suppress the static regions. Specifically, we design the fusion process as below

$$F_i = \Theta_i(F_i' \odot (1 + \texttt{tanh}(O_i))), \quad i = 1,\ 2,\ 3,\ 4, \tag{5.2}$$

where tanh is the hyperbolic tangent non-linearity, and Θ_i is a 1×1 convolution applied for feature fusion. In this step, the tanh non-linearity first scales each element in O_i to the range of (-1, 1), and then the addition of the constant 1 further scales each element to the range of (0, 2). Therefore, $1 + \texttt{tanh}(O_i)$ is a scaling function that may scale up or scale down each element in F_i'. Empirically, as detailed in the ablation studies of Table 5.4, we found that this gating branch improves the performance of the model. The 1×1 convolution Θ_i has the same number of input and output channels, and performs a final projection to embed the re-scaled features into a different high-dimensional space.

5.3.3 *Multi-modal feature decoding*

To fully exploit the information embedded in the feature maps at different granularity levels, we employ a top-down decoding scheme for combining the multi-level, multi-modal feature maps, F_i, $i \in \{1, 2, 3, 4\}$, into a single set of feature maps for mask prediction. The fusion process can be described by the following recursive function

$$\begin{cases} T_4 &= F_4, \\ T_i &= \Psi_i([U(T_{i+1}); F_i]), \quad i = 3,\ 2,\ 1. \end{cases} \tag{5.3}$$

Here '[;]' denotes feature concatenation along the channel dimension, U represents upsampling via bilinear interpolation, and Ψ_i is two 3×3 convolutions connected by batch normalization and ReLU nonlinearity. In words, progressing in the order of F_4, F_3, F_2, and F_1, we repeat an "upsample-concatenate-project" procedure three times in a cascade manner. At each step, we fuse feature maps from the top-down path with feature maps at the current level. The final feature maps, T_1, are projected into two class score maps by a classification layer, which is a 1×1 convolution with stride 1 and 2 output channels. The entire network is trained end-to-end with a cross-entropy loss. During inference, *argmax* along the channel dimension of the score maps are used as the prediction.

5.4 EXPERIMENTS

To evaluate the effectiveness of the proposed HINet, we conduct experimental evaluations on four challenging datasets, including DAVIS-16 [16, 140], DAVIS-17 [30, 140], A2D [79, 139], and J-HMDB [79, 229]. In the following, we first describe the implementation details and evaluation metrics in Section 5.4.1, then we present extensive comparisons with the state-of-the-art methods in Section 5.4.2, and finally discuss several ablation studies in Section 5.4.3.

Method	$\mathcal{J}$(%)	$\mathcal{F}$(%)	$\mathcal{J}\&\mathcal{F}$(%)	year
Khoreva *et al*. [140]	82.8	-	-	2018
RefVOS [83]	71.8	71.0	71.4	2020
HINet (Ours)	**84.4**	**85.3**	**84.8**	-

Table 5.1: Results on DAVIS-2016 validation set. $\mathcal{J}$ and $\mathcal{F}$ are defined in Section 5.4.1. $\mathcal{J}\&\mathcal{F}$ is the mean of $\mathcal{J}$ and $\mathcal{F}$.

5.4.1 *Implementation details and evaluation metrics*

We implement the HINet in PyTorch. We use the SGD optimizer with momentum and weight decay set to 0.9 and 0.0001, respectively. During training, we follow the procedure of previous approaches [79, 82–84, 140] of pre-training first and then fine-tuning. We pre-train our model on YouTubeVOS [31] and then fine-tune it on the experimental datasets for evaluation respectively. During pre-training, we adopt a batch size of 24 and train for 50K iterations. The initial learning rate is set to 0.01 with polynomial learning rate decay. During fine-tuning, we adopt a batch size of 8 and train for 10 epochs on DAVIS-16 and DAVIS-17, and 20 epochs on A2D, with an initial learning rate of 0.001. We use BERT [70] and RAFT [236] for extracting language and optical flow features, respectively. For evaluation metrics, on DAVIS-16 and DAVIS-17, we adopt the official evaluation metrics of mean region similarity $\mathcal{J}$ [16] and mean contour accuracy $\mathcal{F}$ [16]. On A2D and J-HMDB, we adopt the metrics of overall intersection-over-union (oIoU), mean intersection-over-union (mIoU), and precision at five threshold values (P@α where α is a threshold value) [79, 81, 82, 150]. We include detailed definitions for these metrics in Section 5.6.2.

5.4.2 *Comparison with others*

Results on DAVIS-16 and DAVIS-17. In Table 5.1, we evaluate our method against state-of-the-art VOSRE methods on DAVIS-16. Our method outperforms that of Khoreva *et al*. [140] by 1.6 absolute points on the $\mathcal{J}$ metric (only $\mathcal{J}$ is available for their method). We outperform RefVOS [83] by more than 10

Method	1st frame	full video	binary eval.	year
Khoreva *et al*. [140]	39.3	37.1	×	2018
RefVOS [83]	44.5	45.1	×	2020
HINet (Ours)	**50.2**	**47.9**	×	-
URVOS [84]	51.6	–	√	2020
HINet (Ours)	**52.0**	**50.4**	√	-

Table 5.2: $\mathcal{J}\&\mathcal{F}$ on DAVIS-2017 validation set. "1st frame" and "full video" refer to results using 1st-frame expressions and full-video expressions, respectively. For a fair comparison with URVOS | [84], we also report performance under the "binary eval." setting, in which we evaluate each of the multiple referent objects in a video individually.

absolute points on both the $\mathcal{J}$ and $\mathcal{F}$ metrics. As Bellver *et al*. [83] did not report performance on DAVIS-16, we fine-tune their public model (which was pre-trained on RefCOCO [106]) on DAVIS-16 and report the performance.

In Table 5.2, we evaluate our proposed model on the DAVIS-17 validation set. We compare with the method of Khoreva *et al*. [140] and RefVOS [83] under the standard multiple-object segmentation setting. Our method outperforms both methods when leveraging either 1st-frame expressions or full-video expressions. RefVOS [83] is a static segmentation model which segments each frame independently and does not exploit any temporal information. In addition, it only generates multi-modal feature maps from the top-level visual feature maps and does not have a hierarchical feature interaction mechanism. Our performance gain compared with RefVOS [83] validates our approach to integrate motion information and perform multi-level, multi-modal feature fusion. Compared with the method of Khoreva *et al*. [140], our method achieves more than 10% (absolute) improvement in $\mathcal{J}\&\mathcal{F}$, while being end-to-end trainable and enjoys fewer components and significantly simpler training and inference procedures. We compare with URVOS [84] under the binary evaluation setting (see Section 5.6.2). Only performance on the 1st-frame expressions is available for URVOS, and we outperform them by a small margin. We note that our method operates in an online fashion without requiring access to future frames, while URVOS requires the whole video as input for inference.

	Method	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	oIoU	mIoU	Year
A2D	Hu *et al*. [58]	34.8	23.6	13.3	3.3	0.1	47.4	35.0	2016
	Li *et al*. [80]	38.7	29.0	17.5	6.6	0.1	51.5	35.4	2017
	Gavrilyuk *et al*. [79]	50.0	37.6	23.1	9.4	0.4	55.1	42.6	2018
	Wang *et al*. [82]	55.7	45.9	31.9	16.0	2.0	60.1	49.0	2019
	McIntosh *et al*. [150]	52.6	45.0	34.5	20.7	3.6	56.8	46.0	2020
	Wang *et al*. [81]	60.7	52.5	40.5	23.5	4.5	62.3	53.1	2020
	RefVOS [83]	57.8	53.4	45.6	31.1	9.3	67.2	49.7	2020
	Hui *et al*. * [153]	65.4	58.9	49.7	33.3	9.1	66.2	56.1	2021
	Liu *et al*. * [148] (R2D)	59.0	52.7	43.4	28.4	6.8	64.9	51.5	2021
	Liu *et al*. * [148] (I3D)	**65.5**	**59.2**	**50.6**	**34.2**	9.8	65.3	**57.3**	2021
	HINet (Ours)	61.1	55.9	48.6	**34.2**	**12.0**	**67.9**	52.9	-
J-HMDB	Hu *et al*. [58]	63.3	35.0	8.5	0.2	0.0	54.6	52.8	2016
	Li *et al*. [80]	57.8	33.5	10.3	0.6	0.0	52.9	49.1	2017
	Gavrilyuk *et al*. [79]	69.9	46.0	17.3	1.4	0.0	54.1	54.2	2018
	Wang *et al*. [82]	75.6	56.4	28.7	3.4	0.0	57.6	58.4	2019
	McIntosh *et al*. [150]	67.7	51.3	28.3	5.1	0.0	53.5	55.0	2020
	Wang *et al*. [81]	74.2	58.7	31.6	4.7	0.0	55.4	57.6	2020
	RefVOS [83]	73.1	62.0	39.2	8.8	0.0	60.6	56.8	2020
	Hui *et al*. * [153]	78.3	63.9	37.8	7.6	0.0	59.8	60.4	2021
	Liu *et al*. * [148] (I3D)	81.3	65.7	37.1	7.0	0.0	61.6	61.7	2021
	HINet (Ours)	**81.9**	**73.6**	**54.2**	**16.8**	**0.4**	**65.2**	**62.7**	-

Table 5.3: Results on the A2D test set and the entire J-HMDB dataset. The * indicates contemporary work that is published or to be published in 2021.

Results on A2D and J-HMDB. In Table 5.3, we evaluate our method against state-of-the-art language-guided actor-action segmentation methods on A2D and J-HMDB. On the A2D dataset, our method performs the best on 3 out of 7 evaluation metrics. Compared to our model, the contemporary models of Hui *et al*. [153] and Liu *et al*. [148] achieve better precision at the lower IoU thresholds of 0.5, 0.6, and 0.7 and a better mean IoU, but achieves lower precision at the higher IoU thresholds of 0.8 and 0.9. While they demonstrate an overall better object localization ability on the A2D dataset, our model tends to generate finer, more detailed segmentation masks. Our method outperforms the rest of the methods with notable advantages across most of the metrics (except mean IoU on which our performance is lower by 0.2% (absolute) than that of [81]). Both Hui *et al*. [153] and Liu *et al*. [148] employ an I3D [86] backbone which is pre-trained on Kinetics-400 [237] for action classification. When using the same backbone as ours, the method of Liu *et al*. [153] obtains lower performance than ours ("(R2D)" in Table 5.3). Following prior work [79, 81, 82, 150], we test the generalization

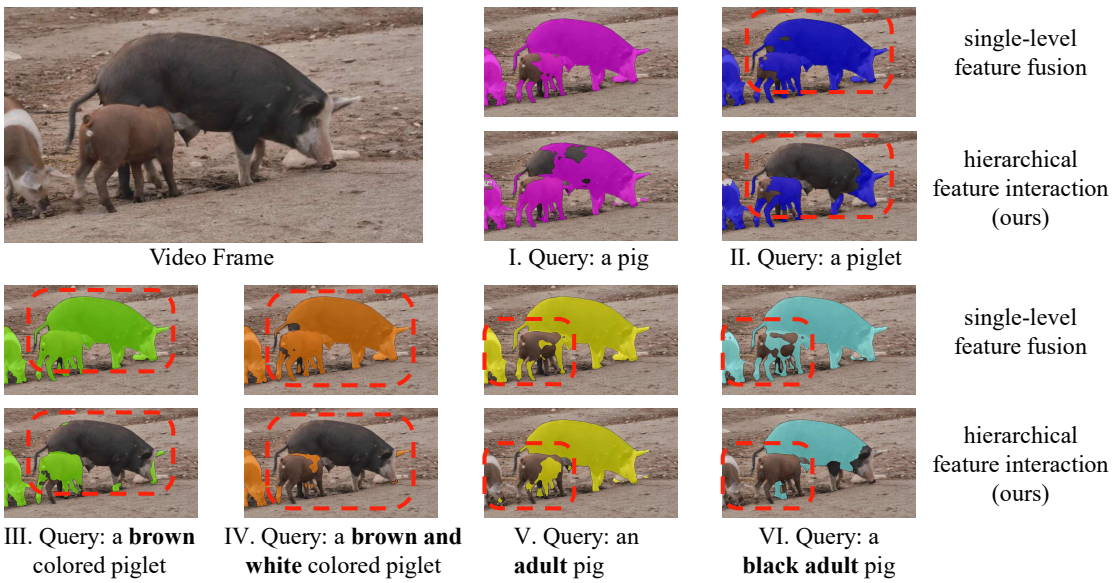


Figure 5.4: Qualitative results on DAVIS-2017. Compared with single-level feature fusion, our hierarchical feature interaction method could generate results with fewer false positives, as indicated in red dashed areas. Moreover, our model could better distinguish different concepts between a "pig" (I) and a "piglet" (II), and learn the concepts of "brown" (III), "white" (IV), "adult" (V), and "black" (VI).

ability of our model on J-HMDB with weights fine-tuned on A2D. As shown in Table 5.3, our method outperforms all other methods on all metrics, mostly by a significant margin. In particular, we improve over the second-best method in precision at the 0.8 and 0.7 thresholds by as large as 8.0 and 15.0 absolute points, respectively. We also achieve non-zero precision at the very hard IoU threshold of 0.9. These results highlight the generalization ability of our model.

5.4.3 *Ablation study*

We conduct several ablations to evaluate the effectiveness of several key design choices, and each corresponds to a row below *Full model* in Table 5.4.

Hierarchical feature interaction. We remove the hierarchical feature interaction mechanism and simply feed the multi-modal feature maps generated from the output of the top-level ASPP module to the final classification layer. This leads to the biggest performance drop (6.3 absolute points) among all ablations, which demonstrates the importance of integrating multiple levels of features that

Method	$\mathcal{J}$(%)	$\mathcal{F}$(%)	$\mathcal{J}\&\mathcal{F}$(%)	$\Delta\mathcal{J}\&\mathcal{F}$
Full model	84.4	85.3	84.8	0.0
Full model w/o. hierarchical feature interaction	78.8	78.1	78.5	-6.3
Full model w/o. optical flow (motion branch)	81.9	82.2	82.0	-2.8
Full model w/o. optical flow (vision branch)	82.0	81.3	81.6	-3.2
Full model (original optical flow)	78.7	80.4	79.5	-5.3
Full model (concatenate)	80.2	79.0	79.6	-5.2
Full model (sum)	82.0	82.0	82.0	-2.8

Table 5.4: Ablation study on the DAVIS-16 validation set. $\Delta\mathcal{J}\&\mathcal{F}$ denotes absolute declines in the average of $\mathcal{J}$ and $\mathcal{F}$.

capture context information at different scales and encode semantics at different levels of complexity. Figure 5.4 shows an example where the hierarchical feature interaction design enables learning of different linguistic attributes and fine-grained semantic concepts. The input query "a pig" generates a mask largely covering all three pigs in the video frame as desired, whereas the input query "a piglet" generates a mask mostly covering only the two small pigs. When more attributes are successively added to describe an object, our model learns to leverage these attributes for more refined referral results. Such trends are easy to see when our predictions are viewed in the orders of "a pig" -> "an adult pig" -> "a black adult pig", and "a piglet" -> "a brown colored piglet" -> "a brown and white colored piglet".

Optical flow. As shown in Table 5.4 ("(motion gating branch)"), removing the motion gating branch leads to a moderate performance drop of 2.8 absolute points. We further evaluate whether it is necessary to include optical flow magnitude in our input to the vision branch ("(vision branch)"). This ablation leads to a decline of 3.2 absolute points in performance. The above experiments show that motion information can be helpful when integrated both at an early stage (to the RGB input) and at a late stage (to the "language-vision" representations). In another ablation experiment, we replace optical flow magnitude with the original optical flow as input to our full model ("(original optical flow)"). The resulting large performance drop of 5.3 absolute points illustrates the importance of choosing optical flow magnitude as the representation. A comparison between

Model	S	H	H + OF (V)	H + OF (M)	H + OF (V&M)	OF
Inference time (ms)	82	88	88 + 30	92 + 30	92 + 30	30
$\mathcal{J}\&\mathcal{F}$(%)	70.2	79.1	82.0	81.6	84.8	–

Table 5.5: Speed-accuracy trade-off analysis. S and H respectively denote single-level and hierarchical feature interaction; H + OF (V) denotes H with optical flow input to the vision branch; H + OF (M) denotes H with a motion gating branch; H + OF (V&M) denotes our full model; OF represents the external optical flow estimation model. For timings in "x+y" format, "x" denotes network inference time and "y" denotes extra optical flow computation time. Time is measured one sample per forward and is averaged on the DAVIS-16 validation set.

this variant and the optical flow-free hierarchical model ("H" in Table 5.5) also shows that using the original optical flow instead of magnitude only brings marginal benefits compared to not using optical flow at all (79.5% vs. 79.1%).

Feature fusion methods. We evaluate two variants which differ from our default model in how they combine the language, vision, and motion features. In the "(concatenate)" variant, we replace multiplication in Equations (3.1) and (3.2) with concatenation followed by a 1×1 convolution for channel reduction. This variant leads to a large performance drop of 5.2 points. In the "(sum)" variant, we replace multiplication in Equations (3.1) and (3.2) with element-wise summation, and the performance decreases by 2.8 absolute points. These results validate our choice of using multiplication for fusing features from different modalities.

Network latency analysis. In Table 5.5, we analyze the latency trade-off when employing our two computationally heavy designs: the hierarchical feature interaction scheme and the incorporation of optical flow inputs. Compared with the single-level feature fusion baseline ("S"), the hierarchical model without optical flow inputs ("H") boosts the performance by 8.9 absolute points with only 6 milliseconds of extra inference time, which corresponds to a 12.7% improvement in accuracy with just 7.3% increase in latency. The efficiency is due to the dilated convolutions adopted in our visual encoder, which makes the lower-level features V_3 and V_2 have the same resolution as the top-level features V_4. In the following, we analyze the performance gains and extra latency of our optical flow variants relative to the optical flow-free baseline "H".

Our full model ("H + OF (V&M)") achieves a 7.2% improvement in accuracy with a 39.8% increase in inference time. Optical flow estimation accounts for most of the inefficiency. When excluding optical flow estimation time, each of the three variants "H + OF (V)", "H + OF (M)", and "H + OF (V&M)" brings 3.7%, 3.2%, and 7.2% performance improvement with just 0.0%, 4.5%, and 4.5% extra latency, respectively.

5.5 DISCUSSION

In this chapter, we have presented the Hierarchical Interaction Network (HINet), a method for addressing the problem of video object segmentation from referring expressions. It integrates language, vision, and motion features in a unified multi-level, multi-modal framework. Unlike previous methods which conduct single-level vision-language feature fusion, HINet exploits the hierarchical structure of the visual encoder and performs feature fusion at multiple levels. This simple network design enables the alignments of visual features of intermediate and low levels with the linguistic features, which we show is helpful for capturing details in the text description as well as preserving spatial details. Extensive experiments on four standard benchmarks demonstrate the advantage of our method compared with the state of the art. In the future, we hope this work can serve as a strong baseline for VOSRE and inspire applications to other related tasks, such as video-based visual question answering.

We'd like to discuss a bit further the use of optical flow in the proposed HINet. As noted in Table 5.5 and the network latency analysis that follows, optical flow introduces a large amount of extra compute and is a big contributor to the inference cost. In addition, the cost of training an optical flow network is not yet considered. One can perhaps argue that the application of optical flow networks *is* to high-level vision tasks such as this one, and therefore using an off-the-shelf model as-is is a fair practice. Still, this detachment introduces problems such as domain shifts and bloated pipelines. In Section 5.6.5 that follows, we

visualize the effects of optical flow on the final predictions, including both positive ones and negative ones, and provide more in-depth analyses. The general conclusion is that optical flow does more good than harm on the results, but in some cases, the predicted flow is not helpful due to errors (which become noise) or that the object being static.

While in HINet, we have used optical flow primarily as saliency signals, they could also be used as correspondences to improve prediction consistency over time. In this regard, however, due to the aforementioned issues with optical flow, the better approach might be similar to the anchor diffusion technique that we have introduced in Chapter 3. There are two ways to implement anchor diffusion in HINet, one is to apply the anchor-diffusion branch and the intra-frame branch to visual features before vision-language feature fusion and the other is to apply them to the vision-language features after integration. Downsampling might be necessary if we implement anchor diffusion in HINet, as GPU memory consumption can be a practical concern. We leave this to future exploration.

5.6 APPENDICES

This section provides implementation details that complement what is described in previous sections of this chapter. Contents are organized as follows: (1) Section 5.6.1 describes the implementation details, (2) Section 5.6.2 discusses the datasets and the evaluation metrics we adopted, (3) Section 5.6.3 compares the performance of our model with that of Khoreva *et al*. [140] on subgroups of DAVIS-16 videos which are organized based on their key characteristics, (4) Section 5.6.4 visualizes the differences in prediction results from several ablated versions of our model, which complement our quantitative ablation study in Section 5.4, (5) Section 5.6.5 discusses and compares the pros and cons of utilizing optical flow in our architecture, (6) Figure 5.7 visualizes our prediction results on several video segments in which the target objects stay stationary, and (7) Figures 5.8 to 5.15 visualize HINet's predictions on all four benchmark datasets.

5.6.1 *Implementation details*

During pre-training, we adopt batch size 24 and 50,000 training iterations with a standard cross-entropy loss. The initial learning rate is 0.01 and follows a "poly" adjustment policy [104], where the initial learning rate is multiplied by $(1 - \frac{iter}{total_iters})^{0.9}$ at each iteration. Network parameters are optimized via stochastic gradient descent with weight decay 0.0001. To improve training speed and memory efficiency, we use mixed precision during training.

The BERT model has a hidden size of 768, which corresponds to c_s in Section 5.3.1 and is initialized with weights from Bellver *et al*. [83]. The official BERT model was pre-trained using two self-supervised tasks, masked language modeling (MLM) and next sentence prediction, with data from BookCorpus [238] (a dataset containing 11,038 unpublished books) and English Wikipedia (excluding lists, tables, and headers). WordPiece [117] with a 30,000 token vocabulary was used as the tokenizer. Bellver *et al*. [83] further fine-tuned the model for one epoch with natural language expressions from the RefCOCO dataset [107] using the MLM objective. Every input sentence is prepended with a special token [CLS] and appended a special token [SEP] indicating the start and the end of the sentence, respectively.

We initialize ResNet-101 [141] with weights pre-trained on ImageNet [215] and randomly initialize all other layers in our model (except BERT). The ASPP module follows the same configuration as in [104]. It presents a total stride of 8 and the output channel number is 256, which corresponds to c_4 in Section 5.3.1. In addition, we set c_1, c_2, and c_3 in Section 5.3.1 to 48, motivated by the practice in [18], which sets a smaller channel number for low-level and intermediate feature maps to prevent their outweighing the importance of the high-level feature maps. We employ RAFT [236], a state-of-the-art deep neural net for optical flow estimation.

Training. It is common practice for referring segmentation methods to first pre-train their models on a large-scale dataset (*e.g*. RefCOCO [106], MSRA [239],

Kinetics [237], YouTubeVOS [31, 84], *etc*.) and then fine-tune them on the evaluation datasets [79, 82–84, 140]. We follow the same practice and pre-train our model on YouTubeVOS and then fine-tune our model on DAVIS-16, DAVIS-17, and A2D for evaluation, respectively. During fine-tuning on DAVIS-16, DAVIS-17, and A2D, we adopt batch size 8, an initial learning rate of 0.001 with the "poly" adjustment policy [104], and weight decay 0.0001, and optimize network parameters via stochastic gradient descent. Following prior work on semi-supervised VOS [231, 232], we only optimize the losses of the hardest 15% pixels. Similar to previous work [80–82], we use class-balanced cross-entropy loss with weight 0.9 for the background class and weight 1.1 for the object class, which we found useful for compensating for the object class as objects are usually much smaller than the background. Finally, for DAVIS-16 and DAVIS-17, we clip the gradient norm to 1.0 during training, which we found important for stabilizing training. The model is trained for 10 epochs on DAVIS-16 and DAVIS-17, and 20 epochs on A2D. During both pre-training and fine-tuning, we resize each frame to the height of 480 pixels and then randomly crop a region of size 480×480 pixels from the frame. Raw predictions are upsampled via bilinear interpolation to the size of the ground-truth masks.

Inference. Each frame is evaluated at its original size. We take the *argmax* along the channel dimension of the score maps as the predicted labels. When evaluating our method on DAVIS-17, we adopt a simple strategy to combine the prediction results of multiple objects in a single frame. Specifically, the predicted mask of an object with a higher ID number overrides that of an object with a lower ID number.

5.6.2 *Datasets and evaluation metrics*

Datasets. We evaluate the performance of our model on four datasets, DAVIS-16 [16, 140], DAVIS-17 [30, 140], A2D [79, 139], and J-HMDB [79, 229]. The DAVIS-16 dataset annotates a single foreground entity for each video and contains 30

videos for training and 20 videos for validation. The DAVIS-17 dataset annotates up to several objects per video and contains 60 videos for training and 30 videos for validation. Khoreva *et al*. [140] augmented the DAVIS-16 and DAVIS-17 datasets with natural language expressions from human annotators for all the original objects. For both DAVIS-16 and DAVIS-17, two expressions are provided based on the annotators' observation in the first frame. For DAVIS-17, two more expressions are provided based on the annotators' observations throughout the whole video. We train our algorithm on the training set and evaluate it on the validation set. The Actor-Action Dataset (A2D) by Xu *et al*. [139] is a benchmark dataset for the task of actor-action video segmentation. There are 43 valid actor-action pairs. This task requires assigning an actor-action class label (or a background label) to each pixel in each frame of the input video. Gavrilyuk *et al*. [79] introduced the task of actor-action segmentation from a sentence, and extended the A2D dataset with natural language descriptions of what action an actor is performing in a video, leading to 6,656 sentences describing 6,656 target objects. There are 3,036 videos for training and 746 videos for testing. We train our model on the training set and evaluate it on the test set. The Joint-annotated Human Motion Data Base (J-HMDB) [229] is a fully annotated dataset for human actions and human poses. There are 928 videos divided into 21 action classes. It provides pixel-level mask annotations in the form of articulated human puppets. Gavrilyuk *et al*. [79] extended this dataset with 928 natural language expressions each describing what the actor is doing in each video. Following previous work [81, 82, 150], we evaluate our method on all 928 videos from this dataset using weights fine-tuned with A2D.

Evaluation metrics. On DAVIS-16 and DAVIS-17, we adopt the official evaluation metrics of mean region similarity $\mathcal{J}$, which is the intersection-over-union of the prediction and ground truth, and mean contour accuracy $\mathcal{F}$, which is the F-measure defined on contour points from the prediction and the ground truth. On DAVIS-17, we report performance under two settings. The first is the standard setting which requires that no pixel be assigned more than one object ID. The

second setting is for a fair comparison with URVOS [84], which did not address the potential overlaps in the predictions of multiple objects (therefore this is an easier setting) and we refer to it as *binary evaluation* in Table 5.2. On A2D and J-HMDB, we adopt the common metrics of overall intersection-over-union (oIoU), mean intersection-over-union (mIoU), and precision at five threshold values. The overall IoU is measured as the ratio between the total intersection area and the total union area of all test samples (each test sample is a language query and a video frame). This metric favors large objects. The mean IoU is the IoU between the prediction and ground truth averaged across all test samples. This metric treats large and small objects equally. The precision metric measures the percentage of test samples that passes a certain IoU threshold. We evaluate precision at the common IoU thresholds of 0.5, 0.6, 0.7, 0.8, and 0.9.

5.6.3 *Attribute-based performance on DAVIS-16*

Attribute	AC	LR	SV	SC	CS	DB	BC	FM	MB	DEF	OCC
Khoreva *et al*. [140]	80.1	79.0	74.4	**77.6**	85.7	66.4	85.0	77.7	78.1	**84.3**	80.1
RefVOS(re-implement) [83]	68.3	67.2	67.2	64.5	79.7	48.5	89.3	63.7	66.5	71.2	68.2
Ours	**85.1**	**88.3**	**83.3**	77.1	**92.3**	**69.2**	**90.3**	**82.6**	**80.7**	83.7	**84.7**

Table 5.6: Attribute-based results ($\mathcal{J}$) on the DAVIS-16 validation set. AC: appearance change. LR: low resolution. SV: scale variation. SC: shape complexity. CS: camera shake. DB: dynamic background. BC: background clutter. FM: fast motion. MB: motion blur. DEF: deformation. OCC: occlusions.

In Table 5.6, we provide an analysis of our method when evaluated on DAVIS-16 videos assigned to different attribute groups. Our method achieves the best performance in most attribute groups and is robust in a wide variety of challenging scenarios.

5.6.4 *Ablative qualitative analysis*

In Figure 5.5, we compare the different prediction results generated by models that employ only single-level feature fusion and our hierarchical feature interac-

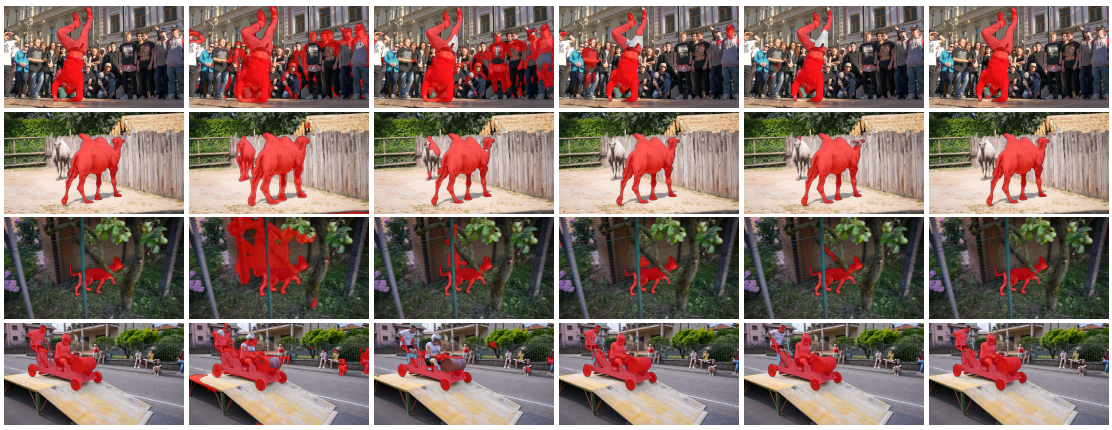


Figure 5.5: We visualize the effects of our key contributions on examples from DAVIS-16. From left to right, each column visualizes masks from the ground truth, model that employs single-level feature fusion without utilizing any optical flow input, models that employ hierarchical feature interaction only between language and vision features without any optical flow input, with optical flow input only to the vision branch, with optical flow input only to the motion gating branch, and with optical flow input to both branches, respectively.

tion scheme (both models do not utilize any optical flow input); and then we demonstrate the effects of incorporating optical flow input to the vision branch, to the motion gating branch, and to both branches, respectively, under our hierarchical feature interaction scheme. It can be seen that hierarchical feature interaction even without optical flow input generates fewer false positives and more accurate masks compared to its single-level feature fusion counterpart. Moreover, supplying optical flow input either to the vision branch or to the motion gating branch helps the model focus on the correct target object, which demonstrates the benefit of utilizing motion information. Finally, incorporating optical flow in both branches yield the best results, which further shows that our two ways of utilizing optical flow complement each other and are both necessary for achieving the best results.

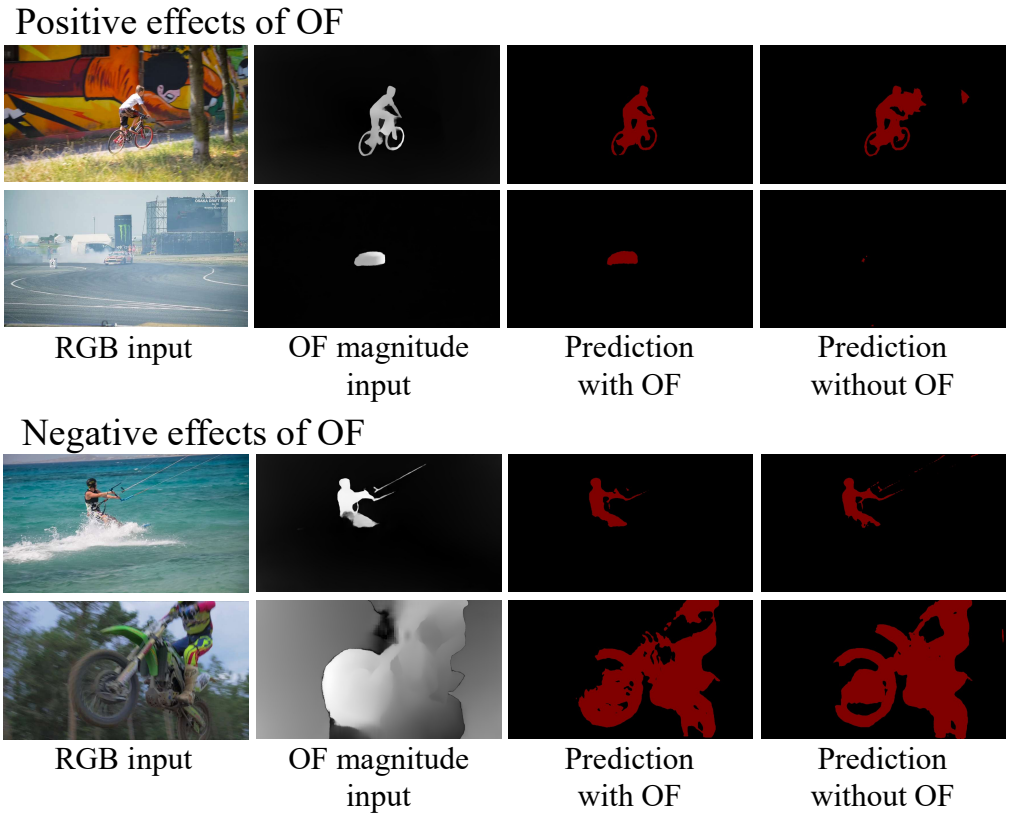


Figure 5.6: Incorporating optical flow for prediction have both positive and negative effects. However, as discussed in the text of this section as well as demonstrated in Table 5.4, the benefits of utilizing optical flow far outweighs the drawbacks.

5.6.5 *The effects of optical flow*

An important question is that if we employ optical flow for highlighting moving objects, then how much do poor optical flow predictions and static target objects affect our predictions? We discuss the positives and the negatives of using optical flow in our architecture with a few examples in Figure 5.6. First of all, as can be seen in the first two rows under "Positive effects of OF," the main benefit of using optical flow is to complement language information for more accurate object localization. In many cases, as language referral is an extremely challenging problem, the expression itself often generates false positive or false negative predictions. These are the cases where optical flow magnitude (computed after subtracting the mean vector) comes into play and helps correct those false positives or false negatives.

In some cases (the two rows under "Negative effects of OF"), incorporating OF produces slight negative effects. In the first case, motion of waves causes the model to segment out water near the legs of the referent person. In the second case, due to camera rotations and the target being near the camera, subtracting the mean vector does not produce a decent separation between the moving target and the static background. The optical flow magnitude image is uninformative and includes many distracting signals. Even in this case, the prediction result from incorporating optical flow is only slightly worse than that from the RGB-only model, missing only small detailed parts of the motorbike and the biker. This shows that when optical flow signals are of poor quality, our model can still rely on language to produce decent predictions.

Finally, we want to mention that the overall benefits of optical flow far outweigh the negative effects. There are two main reasons. First, consistent with the pattern shown in these four examples, optical flow tends to make *big* improvements by correcting false positives or negatives, and only introduces *small* errors in the cases where they are harmful. Second, in all four benchmarks, the cases where optical flow plays a positive role far outnumber those where it negatively affects the prediction results. This is due to the fact that the queried target objects in videos are more often moving independently from the background, making it easy to separate their motion from the background by subtracting the mean vector from the flow field.

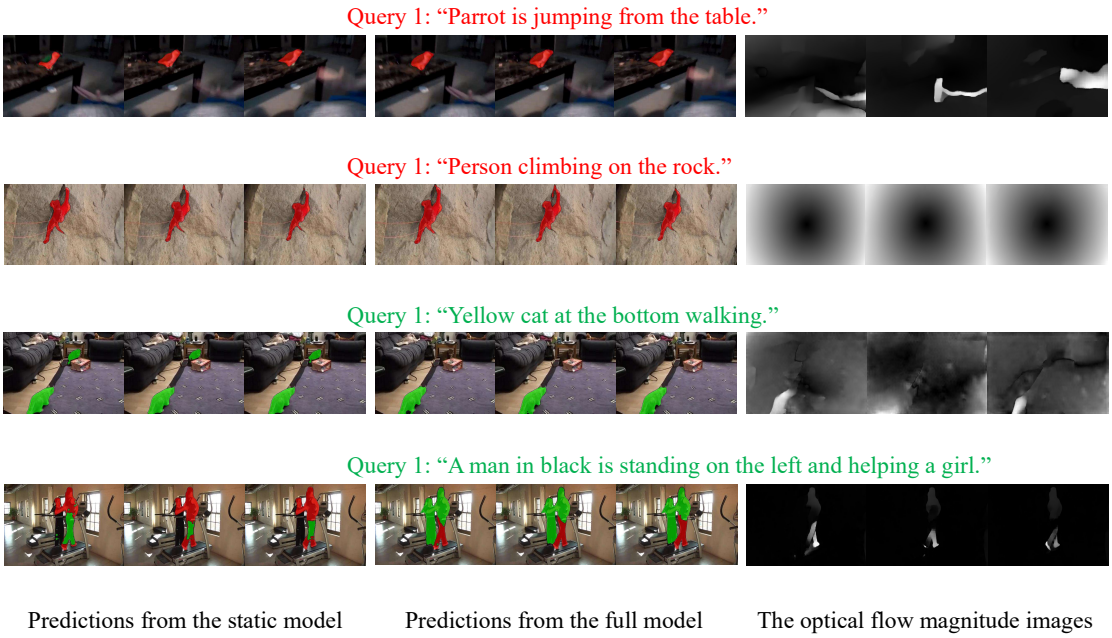


Figure 5.7: Predictions on a video segment in which the target object is stationary. We include predictions from the static model which only employs language as guidance for comparison with predictions from our full model. When motion signals about the queried object are missing, our model can still leverage language to locate the object and generates similar or better masks compared with its static counterpart.



Figure 5.8: Segmentation results on the DAVIS-16 validation set.



Figure 5.9: Segmentation results on the DAVIS-16 validation set.



Figure 5.10: Segmentation results on the DAVIS-17 validation set, excluding repetitive videos also contained in the DAVIS-16 validation set.



Figure 5.11: Segmentation results on the DAVIS-17 validation set, excluding repetitive videos also contained in the DAVIS-16 validation set.



Figure 5.12: Segmentation results on challenging videos from the A2D test set.

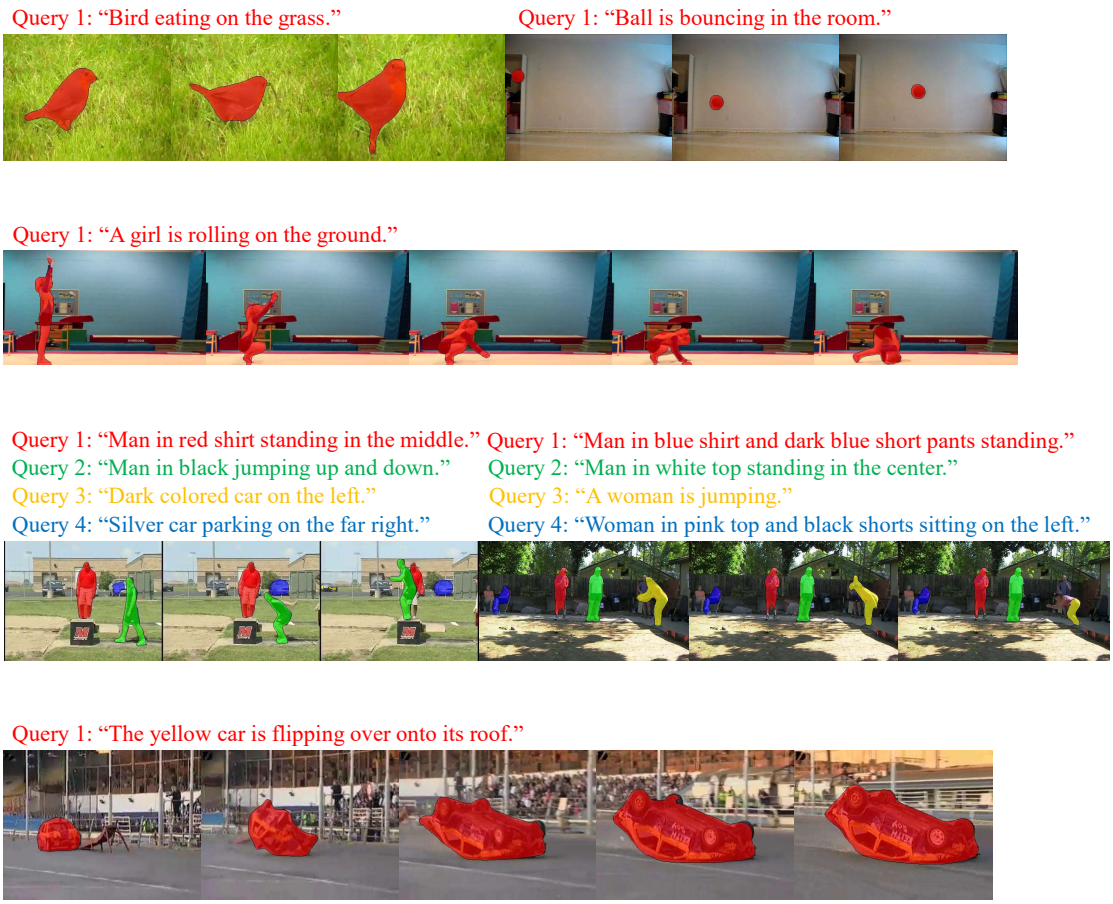


Figure 5.13: Segmentation results on challenging videos from the A2D test set.

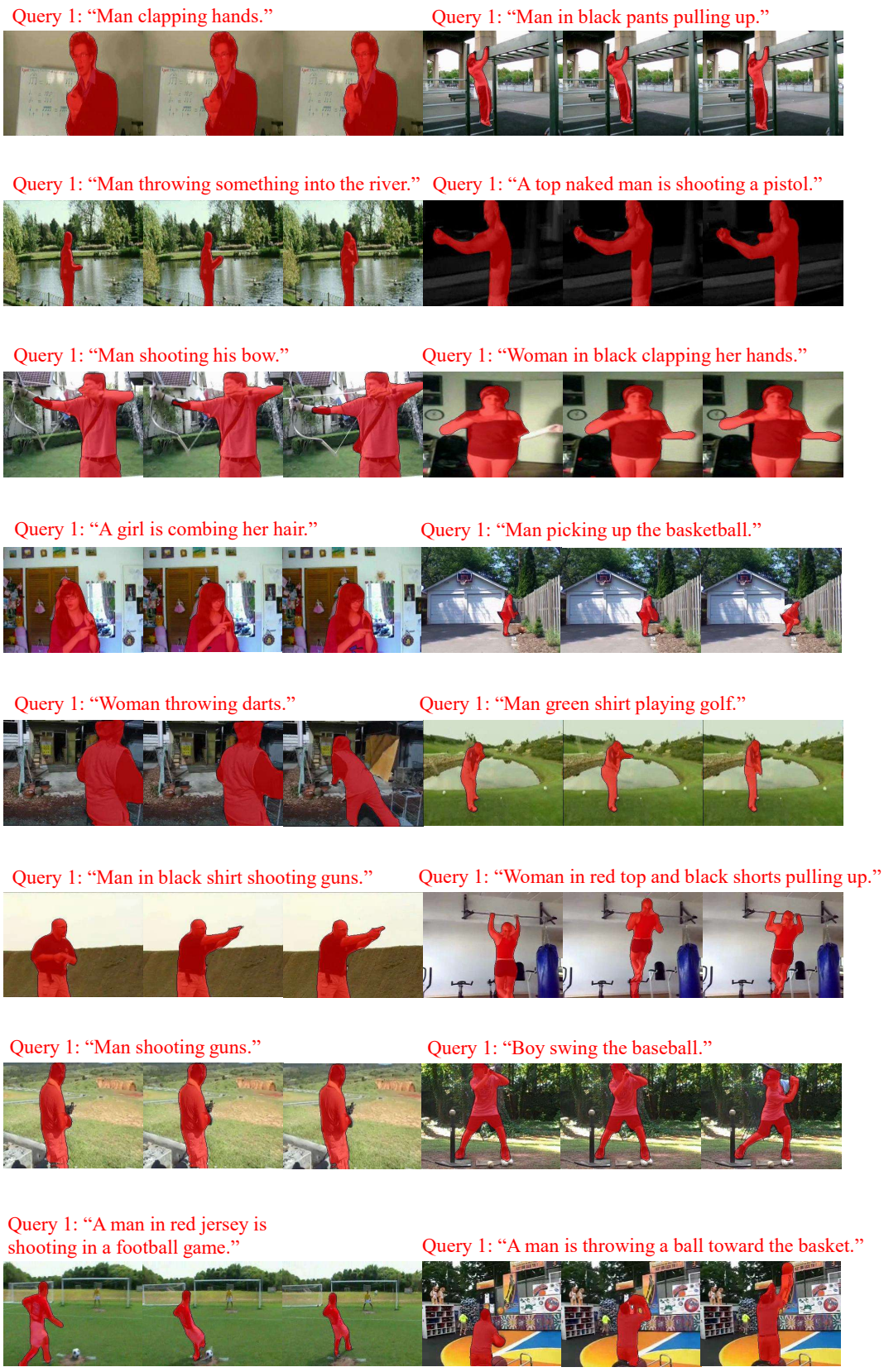


Figure 5.14: Segmentation results on challenging videos from the J-HMDB dataset.

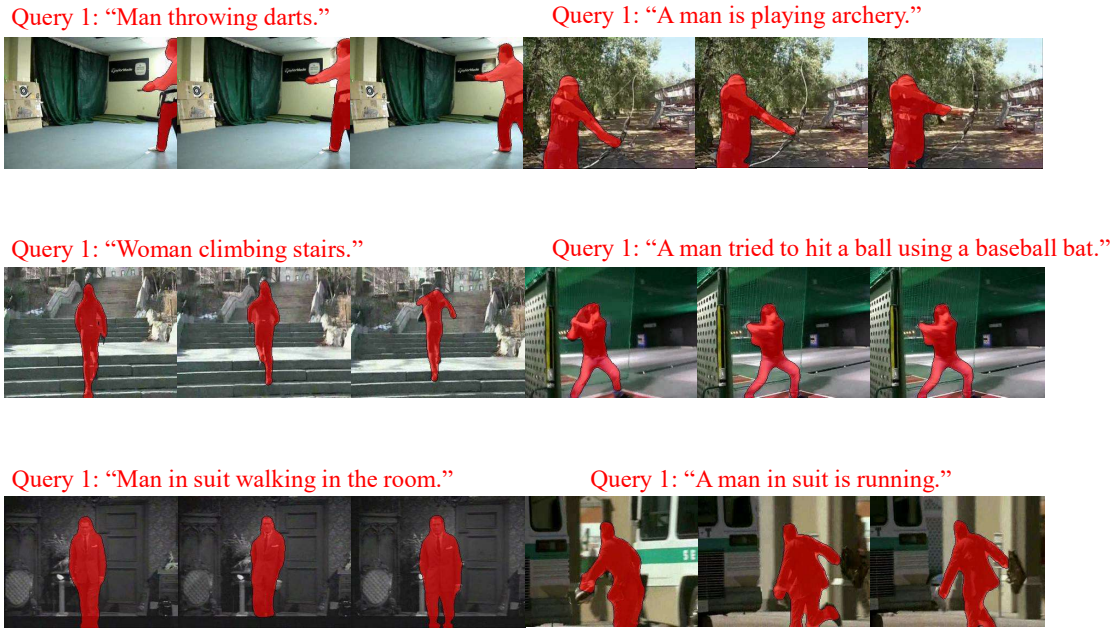


Figure 5.15: Segmentation results on challenging videos from the J-HMDB dataset.

5.7 STATEMENT OF AUTHORSHIP

The paper [91] from which this chapter originates is multi-authored, as is common with papers in science-based subjects. We specify the extent of contribution of each author as follows.

- Zhao Yang participated in the brainstorming of ideas, wrote code, ran experiments, and prepared the manuscript.
- Yansong Tang participated in the brainstorming of ideas and contributed to the preparation of the manuscript.
- Luca Bertinetto participated in the brainstorming of ideas and reviewed the manuscript.
- Hengshuang Zhao participated in the brainstorming of ideas and contributed to the preparation of the manuscript.
- Philip HS Torr organized the project, participated in the brainstorming of ideas, and reviewed the manuscript.

6

CONCLUSION

The research of visual object segmentation is often carried out on static images based on a set of pre-defined semantic classes. Although this approach is effective, it does not allow the segmentation of arbitrary objects or the use of temporal information for segmentation, which can hurt algorithms' prospects of real-world deployment. Motivated by the relevance of overcoming such limitations, in this thesis, we have set out to explore the potential of exploiting linguistic and temporal cues for addressing visual object segmentation. By investigating three important computer vision tasks, *i.e.*, automatic video object segmentation, image segmentation from referring expressions, and video object segmentation from referring expressions, we arrived at several novel architectural designs of deep neural networks that show great promise for our purpose.

In Chapter 3, we demonstrated that object segmentation accuracy can be significantly improved when an algorithm takes account of temporal information in a video. This is achieved via a technique that learns how to effectively aggregate context from a reference frame for helping the segmentation of each frame in a video. It establishes pixel correspondences between a reference (the "anchor") frame and the current (the "target") one, and propagates features between these two frames. This approach is particularly helpful in frames where the appearance of the target object is "atypical," *e.g.*, blurred, enlarged, shrunk, partially occluded, *etc*. In these cases, representations of the object can be retrieved from the reference frame to aid segmentation. For this reason, the proposed algorithm is able to produce much better results in videos compared

to a static foreground-background segmentation model. At the time of publication, the proposed algorithm obtained the highest accuracy on two popular benchmarks for unsupervised video object segmentation, while performing surprisingly well against the state-of-the-art methods in semi-supervised video object segmentation and video saliency.

The setup under which we studied object segmentation in Chapter 3 assumes that the target objects are salient (explained in Section 1.2.1 and Section 2.1). While this simple assumption has allowed us to better focus on the problem of temporal dependency modeling, it is ambiguous and still impractical in many real-world situations. To address those problems, in Chapters 4 and 5 we studied language-guided visual object segmentation in images and videos, respectively.

In Chapter 4, we introduced LAVT, a language-aware vision Transformer for addressing the problem of referring image segmentation. Specifically, we have shown that the learning of cross-modal feature alignment, central to this problem, can be addressed via the early fusion of linguistic and visual features in intermediate layers of a vision Transformer network. This approach alleviates the need for additional, computationally heavy cross-modal feature fusion modules, and segmentation can be harvested with standard upsampling convolution layers. The proposed method significantly improved results (measured by IoU) reported for the datasets of RefCOCO (from 65% to 72%), RefCOCO+ (from 55% to 62%), G-Ref-UMD (from 54% to 61%), and G-Ref-Google (from 51% to 60%).

In Chapter 5, we presented a unified network that jointly models visual, linguistic, and temporal cues for addressing video object segmentation from referring expressions. The core idea of our approach is to leverage the feature pyramid inherently existing in a convolutional neural network (CNN), and integrate both optical flow features and linguistic features with visual features at different levels of the CNN. This approach establishes cross-modal alignments at multiple scales and levels of semantic complexity, and highlights features of moving objects in the multi-level, multi-modal feature maps. This leads to more accurate pixel-wise referrals compared to previous methods that performed

single-level feature fusion. Moreover, temporal information can be exploited via optical flow rather than 3D CNNs, which alleviates the spatial misalignment problem faced by previous methods. At the time of publication, the proposed method obtained state-of-the-art results on four popular benchmarks.

6.1 FUTURE WORK

Generalization via large-scale vision-language pre-training. Our interest in object segmentation from referring expressions stems from a desire to develop an algorithm that is capable of segmenting out *any* objects from an image or video, as opposed to just objects of certain categories. In practice, an algorithm would only be able to do that if it could understand *any* referring expressions. Currently, we adopt a language model that is built on very large corpora (see Section 2.2), and this can help our model generalize to unseen words (*e.g.*, "vehicle") at test time that are semantically similar to words having been seen during training (*e.g.*, "car"). However, the model would not be able to generalize to completely new words that are unrelated to words seen during training, as the cross-modal alignment function would have difficulty to measure the similarity between the strange words and regions in the image or video.

One possible way to address this issue is to leverage both a vision encoder and a language encoder (or a hybrid multi-modal encoder) that have been pre-trained on a large amount of paired image-text data. One example of such pre-trained models is CLIP [198], which learns to match a piece of text to an image via contrastive learning [114, 115, 207, 208], among many other examples [200, 240–243]. Aside from using the pre-trained weights for network initialization, we think that there are two other promising ways of exploiting them. The first is to extract supervision signals from the pre-trained model for supervising the learning of segmentation, and there is a rich set of tools from the literature of weakly supervised segmentation [244] that we can explore. To

this end, the general idea of employing analytical saliency maps (*e.g.*, Grad-CAM [245]) as pseudo-labels and adopting an iterative learning strategy may be a good starting point. The second is to compute several similarity maps between the input text and the image, and use them as additional channels for guiding segmentation. The expectation is that the established cross-modal alignments in the pre-trained model can provide strong localization signals for the target object.

Robust object segmentation in the presence of ambiguity. In scenarios where referring image segmentation may be useful, such as image editing and communicating to a household robot, the verbal expressions from the user do not have to be entirely correct, in the sense that the user may mistake a color or an action, or falsely identify an object that does not exist. What does an algorithm do in such cases? Currently, as the problem setup always assumes that the phrase is unambiguous and correct, algorithms try to highlight at least some regions in the image or video, and this is probably not the more useful behavior or what is expected by the user.

To cope with ambiguity or mistakes in the expressions, we should devote efforts to the construction of datasets, evaluation metrics, and algorithms, probably in that order. And this is one potential way of doing it. In addition to the unambiguous and correct expressions that exist in current benchmark datasets, we can add two more sections of expression annotations. The first contains ambiguous expressions, which have minor inaccuracies in the description but we still consider good enough to lead the algorithm to the correct target. This essentially asks the algorithm to be ambiguity-tolerant to a reasonable degree, and we can code the types of ambiguities we think are reasonable into the phrases by construction. The evaluation for this section of phrases can adopt both the existing overall IoU and mean IoU metrics, as well as new metrics such as mean average precision (mAP) [195] and mean absolute error (MAE) [11], which take into consideration the algorithm's "confidence" in the predictions. The second section may contain expressions that are false, *i.e.*, being totally wrong and the only correct prediction from which should be an indication that the object

does not exist. The evaluation for this section of phrases can potentially be based on the false positive rate or the mean IoU, where a prediction is false positive if it is a non-empty mask, and the IoU is measured between the prediction and the entire image. In either case, a lower number indicates a better performance.

Object segmentation's connection with image generation/inpainting and referring expression generation. Much like how well-constructed multi-task learning frameworks or soundly selected auxiliary losses can in general benefit the learning of a task, there is reason to think that object segmentation from language can benefit from a joint learning framework that incorporates one or several of the tasks of image generation [219, 246–248], image inpainting [249–251], or referring expression generation [106, 109, 252]. The hope is that solving these tasks induces the learning of complementary features beneficial to each other.

For instance, by masking out a region containing an object and providing a text description for the object, we can require the algorithm to fill in the missing region according to the text description. A network trained for this task would learn an appearance model of the object that is conditioned on both the text input and the image content. In principle, this appearance model encodes the most essential characteristics of the object, and thus if used for referring image segmentation, may help with the learning of cross-modal feature alignment. Similar reasoning can be done in the reverse direction and with the task of image generation from language. As another example, referring expression generation (REG), *i.e.*, the task of generating an unambiguous description for a given image region, is the opposite of referring image segmentation (RIS). One possibility is to train a single model that addresses both tasks simultaneously, and another possibility is to make the two tasks mutually dependent on each other, *i.e.*, using the predicted mask from the RIS model as input to the REG model, and vice versa.

Self-supervised learning of image and video representations for object segmentation. Much of the discussion on object segmentation in this thesis concerns the fully supervised setting, where ground-truth masks are available for large

amounts of data and the learning of deep representations of objects depends on the supervision from these masks. However, the fully supervised approach is not truly scalable due to the high labeling costs. As a result, many recent studies instead develop techniques aimed at self-supervised learning, which take advantage of unlabeled data for learning pixel-wise deep representations that are helpful for object segmentation.

Among methods that are designed to operate on static images [253–258], Ji *et al*. [255] and Ouali *et al*. [256] suggest to maximize the agreement between cluster assignments of two transformed versions ("views") of a pixel, Gansbeke *et al*. [258] propose to treat unsupervised saliency masks as clusters of pixels and maximize pairwise similarities within each cluster and pairwise dissimilarities across clusters, and Voynov *et al*. [257] propose to segment an image into foreground and background regions using an unsupervised generative adversarial network. In these methods, deriving segmentation from the representations typically involves an additional step of clustering, linear classification, or thresholding.

Among methods that exploit temporal cues in a video [180, 259–262], Vondrick *et al*. [259] leverage colorization as a proxy task for learning pixel correspondences across frames, while Jabri *et al*. [180] formulate space-time correspondence learning as a contrastive random walk that exploits cycle-consistency. The quality of the learned representations is typically evaluated in label propagation tasks, such as semi-supervised video object segmentation (desribed in Section 2.1).

Although great progress has been made with regard to the self-supervised learning of pixel-level object representations, making predictions from such representations still requires prior knowledge about object categories. Perhaps an interesting direction to head into is that of novel category discovery [263, 264], currently formulated for the task of image classification but should be pertinent to the problem of object segmentation as well.

Segmentation based on "intentions." In the end, maybe what we want, for a system that demonstrates some level of intelligence and has segmentation as part of its functionality, is to be able to know first what information needs to be extracted as the result of the segmentation, and then segments in a way that helps it get this information. The motivation goes back to the examples in Figure 1.1, where we have shown that how we want a segmentation to be done is inseparable from the intended use for that segmentation.

Referring image segmentation, which we studied in this thesis, is one step in this direction, as language is the most typical and convenient way for us to communicate our thoughts. But language is a formality, which is necessary for communication, but superfluous to the operations of a system. For instance, before we do something, we don't talk to ourselves and say what to do, instead, we do it.

A generalized notion of segmentation from intentions perhaps can be investigated for an agent learning to interact with its environment, such as the ones trained via reinforcement learning [265–267]. Given an agent and its first-person/third-person view of the world, we could train the agent to segment out areas important to its next steps, which would make the segmentation dependent on the agent's policy state. One potential way of evaluation is to observe, in new episodes of the environment, whether the segmentation produced by the agent can be explained by its strategies, and vice versa. Another possible formulation of this problem is the following. Given an image or a video, we require the algorithm to output a segmentation along with a description for each segment, explaining the content of the segment. Then for each image (or frame in a video), there should be multiple ground-truth segmentations annotated with appropriate language descriptions. However, in this formulation, how we might represent the "intention" which should be an input to the algorithm is unclear.

Much more thinking and investigation need to be done before this idea can be developed into a well-founded research direction.

REFERENCES

[1] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017 (cited on page 2).

[2] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016 (cited on page 2).

[3] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & virtual environments* 6.4, 1997, pp. 355–385 (cited on page 2).

[4] Paul Viola and Michael J Jones. Robust real-time face detection. In *IJCV*, 2004 (cited on page 2).

[5] David F Ferraiolo, John F Barkley, and D Richard Kuhn. A role-based access control model and reference implementation within a corporate intranet. In *ACM TISSEC*, 1999 (cited on page 2).

[6] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015 (cited on page 2).

[7] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017 (cited on page 2).

[8] Beste F Yuksel, Pooyan Fazli, Umang Mathur, Vaishali Bisht, Soo Jung Kim, Joshua Junhee Lee, Seung Jung Jin, Yue-Ting Siu, Joshua A Miele, and Ilmi Yoon. Human-in-the-loop machine learning to increase video accessibility for visually impaired and blind users. In *ACM DIS*, 2020 (cited on page 2).

[9] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. In *TPAMI*, 2012 (cited on page 4).

[10] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. In *TPAMI*, 2010 (cited on page 4).

[11] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. In *TPAMI*, 2014 (cited on pages 4, 129).

[12] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006 (cited on page 4).

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015 (cited on pages 4, 5, 8, 19, 20, 67).

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017 (cited on pages 4, 5, 8, 19, 24, 96).

[15] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019 (cited on pages 4, 8).

[16] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016 (cited on pages 4, 6, 7, 15, 17, 27, 33, 44, 45, 56, 92, 94, 96, 100, 101, 110).

[17] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *CVPR*, 2006 (cited on pages 4, 5).

[18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018 (cited on pages 5, 46, 94, 109).

[19] Philip HS Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 356.1740, 1998, pp. 1321–1340 (cited on page 5).

[20] M Pawan Kumar, Philip HS Torr, and Andrew Zisserman. Learning layered motion segmentations of video. In *IJCV*, 2008 (cited on page 5).

[21] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *ACCV*, 2020 (cited on page 5).

[22] Pia Bideau. Motion Segmentation - Segmentation of Independently Moving Objects in Video. PhD thesis. University of Massachusetts Amherst, 2020 (cited on page 5).

[23] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. In *IJCV*, 2015 (cited on page 5).

[24] 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 *ECCV*, 2014 (cited on page 5).

[25] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019 (cited on page 5).

[26] `https://www.v7labs.com/blog/panoptic-segmentation-guide`. Online; accessed September 2022 (cited on page 5).

[27] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. In *TPAMI*, 2013 (cited on pages 6, 7, 17, 33, 36, 45, 59, 96).

[28] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. In *TIP*, 2015 (cited on pages 6, 7, 17, 33, 36, 45, 49, 50, 59, 96).

[29] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013 (cited on pages 6, 7, 17).

[30] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. In *CVPR Workshops*, 2017 (cited on pages 6, 27, 94, 95, 100, 110).

[31] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *ECCV*, 2018 (cited on pages 6, 27, 92, 101, 110).

[32] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 DAVIS challenge on VOS: Unsupervised multi-object segmentation. In *CVPR Workshops*, 2019 (cited on pages 6, 33, 92, 96).

[33] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *CoRL*, 2017 (cited on pages 7, 33).

[34] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The kitti vision benchmark suite. In *CVPR*, 2012 (cited on pages 7, 33).

[35] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. In *IJRR*, 2017 (cited on pages 7, 33).

[36] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *ICRA*, 2016 (cited on pages 7, 33, 92).

[37] Roberto Vezzani, Davide Baltieri, and Rita Cucchiara. People reidentification in surveillance and forensics: A survey. In *CSUR*, 2013 (cited on pages 7, 33, 92).

[38] Federico Perazzi. Video object segmentation. PhD thesis. ETH Zurich, 2017 (cited on pages 7, 33, 92).

[39] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper ConvLSTM for video salient object detection. In *ECCV*, 2018 (cited on pages 7, 18, 33, 37, 45, 47, 48, 57, 96).

[40] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017 (cited on pages 7, 17, 18, 33, 35, 37, 45, 48, 57, 96).

[41] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. SegFlow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017 (cited on pages 7, 17, 33, 37, 48, 57).

[42] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, 2017 (cited on pages 7, 17, 33, 48, 57).

[43] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017 (cited on pages 7, 17, 33, 48, 57).

[44] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C-C Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *CVPR*, 2018 (cited on pages 7, 8, 17, 18, 33, 35, 37, 38, 45).

[45] Siyang Li, Bryan Seybold, Alexey Vorobyov, Xuejing Lei, and C.-C. Jay Kuo. Unsupervised video object segmentation with motion-based bilateral networks. In *ECCV*, 2018 (cited on pages 7, 17, 18, 33, 37, 38, 45, 96).

[46] Mennatullah Siam, Chen Jiang, Steven Lu, Laura Petrich, Mahmoud Gamal, Mohamed Elhoseiny, and Martin Jagersand. Video object segmentation using teacher-student adaptation in a human robot interaction (HRI) setting. In *ICRA*, 2019 (cited on pages 7, 17, 33, 36, 37, 45, 47, 48, 57).

[47] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, 2017 (cited on pages 7, 17, 33, 36, 48, 57, 96).

[48] Ilya Sutskever. Training recurrent neural networks. PhD thesis. University of Toronto, 2013 (cited on pages 7, 35, 48).

[49] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. In *Neural Computation*, 1997 (cited on pages 7, 18, 37, 67, 86).

[50] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014 (cited on pages 7, 17, 22, 37).

[51] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. In *IEEE Transactions on Neural Networks*, 1994 (cited on pages 7, 18, 35, 37, 48).

[52] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013 (cited on pages 7, 18, 35, 37, 48).

[53] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014 (cited on pages 7, 19, 22, 35).

[54] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018 (cited on pages 8, 16–18, 35–37, 48, 96).

[55] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018 (cited on pages 8, 16–18, 35, 37, 48, 96).

[56] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017 (cited on pages 8, 15, 16, 18, 35–37, 46, 95).

[57] Ming-Ming Cheng, Shuai Zheng, Wen-Yan Lin, Vibhav Vineet, Paul Sturgess, Nigel Crook, Niloy J. Mitra, and Philip Torr. ImageSpirit: Verbal guided image parsing. In *TOG*, 2014 (cited on pages 8, 19, 65).

[58] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016 (cited on pages 8, 19–21, 65, 67, 71, 92, 95, 103).

[59] 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 MM*, 2020 (cited on pages 8, 23, 24, 71, 77, 79, 80).

[60] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language Transformer and query generation for referring segmentation. In *ICCV*, 2021 (cited on pages 8, 9, 24, 65, 66, 68, 77, 81).

[61] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *ICCV*, 2017 (cited on pages 8, 21, 65, 67, 95).

[62] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *ECCV*, 2018 (cited on pages 8, 21, 71, 77).

[63] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *ECCV*, 2018 (cited on pages 8, 23, 65, 67, 71, 95).

[64] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *CVPR*, 2019 (cited on pages 8, 23, 24, 67, 71, 77).

[65] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *ICCV*, 2019 (cited on pages 8, 21, 23, 65, 67, 77, 95).

[66] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In *CVPR*, 2020 (cited on pages 8, 23, 24, 29, 65, 67, 71, 77, 79, 80, 95).

[67] 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 *CVPR*, 2020 (cited on pages 8, 23, 24, 29, 65, 68, 77, 95).

[68] 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 *ECCV*, 2020 (cited on pages 8, 23, 24, 65, 68, 77).

[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017 (cited on pages 9, 21, 23, 24, 29, 39, 41, 54, 65, 68, 72, 73, 75).

[70] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *NAACL*, 2019 (cited on pages 9, 21, 23, 25, 67, 68, 86, 98, 101).

[71] Li Muchen and Sigal Leonid. Referring Transformer: A one-step approach to multi-task visual grounding. In *NeurIPS*, 2021 (cited on pages 9, 24, 25).

[72] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, 2019 (cited on pages 9, 68).

[73] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019 (cited on pages 9, 68).

[74] 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 *ICLR*, 2021 (cited on pages 9, 20, 23, 68).

[75] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with Transformers. In *ECCV*, 2020 (cited on pages 9, 25, 55, 68).

[76] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for end-to-end object detection. In *ICLR*, 2021 (cited on pages 9, 55, 68).

[77] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with Transformers. In *CVPR*, 2021 (cited on pages 9, 20, 68).

[78] 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 *ICCV*, 2021 (cited on pages 9, 20, 23, 55, 66, 68, 69, 71, 75, 84).

[79] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *CVPR*, 2018 (cited on pages 10, 26, 28, 29, 92, 94–96, 100, 101, 103, 110, 111).

[80] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees GM Snoek, and Arnold WM Smeulders. Tracking by natural language specification. In *CVPR*, 2017 (cited on pages 10, 92, 95, 96, 103, 110).

[81] Hao Wang, Cheng Deng, Fan Ma, and Yi Yang. Context modulated dynamic networks for actor and action video segmentation with language queries. In *AAAI*, 2020 (cited on pages 10, 28, 29, 92, 94–96, 101, 103, 110, 111).

[82] Hao Wang, Cheng Deng, Junchi Yan, and Dacheng Tao. Asymmetric cross-guided attention network for actor and action video segmentation from natural language query. In *ICCV*, 2019 (cited on pages 10, 29, 92, 94–96, 101, 103, 110, 111).

[83] Miriam Bellver, Carles Ventura, Carina Silberer, Ioannis Kazakos, Jordi Torres, and Xavier Giro-i-Nieto. RefVOS: A closer look at referring expressions for video object segmentation. *arXiv preprint arXiv:2010.00263*, 2020 (cited on pages 10, 67, 92, 95, 101–103, 109, 110, 112).

[84] Seonguk Seo, Joon-Young Lee, and Bohyung Han. URVOS: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, 2020 (cited on pages 10, 27, 92, 94–96, 101, 102, 110, 112).

[85] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015 (cited on pages 11, 28).

[86] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017 (cited on pages 11, 28, 92, 103).

[87] Tianrui Hui, Shaofei Huang, Si Liu, Zihan Ding, Guanbin Li, Wenguan Wang, Jizhong Han, and Fei Wang. Collaborative spatial-temporal modeling for language-queried video actor segmentation. In *CVPR*, 2021 (cited on page 11).

[88] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal Transformers. In *CVPR*, 2022 (cited on pages 11, 30, 31).

[89] Zhao Yang, Qiang Wang, Luca Bertinetto, Song Bai, Weiming Hu, and Philip H.S. Torr. Anchor diffusion for unsupervised video object segmentation. In *ICCV*, 2019 (cited on pages 11, 62, 96).

[90] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip H. S. Torr. LAVT: Language-aware vision Transformer for referring image segmentation. In *CVPR*, 2022 (cited on pages 12, 90).

[91] Zhao Yang, Yansong Tang, Luca Bertinetto, Hengshuang Zhao, and Philip H.S. Torr. Hierarchical interaction network for video object segmentation from referring expressions. In *BMVC*, 2021 (cited on pages 13, 78, 125).

[92] Sergi Caelles, Kevis-Kokitsi. Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017 (cited on pages 15, 36, 48, 95).

[93] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for object tracking. In *CVPR Workshops*, 2017 (cited on pages 15, 16, 36).

[94] Xiaoxiao Li, Yuankai Qi, Zhe Wang, Kai Chen, Ziwei Liu, Jianping Shi, Ping Luo, Chen Change Loy, and Xiaoou Tang. Video object segmentation with re-identification. In *CVPR Workshops*, 2017 (cited on pages 15, 16, 36).

[95] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. PReMVOS: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018 (cited on pages 15, 16, 36, 48, 95).

[96] Kevis-Kokitsi. Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. In *TPAMI*, 2018 (cited on pages 15, 16, 36, 95).

[97] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017 (cited on pages 15, 16, 36, 48, 95).

[98] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018 (cited on pages 16, 17, 36).

[99] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, 2018 (cited on pages 16, 17).

[100] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018 (cited on pages 16, 17).

[101] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for video object segmentation. In *IJCV*, 2019 (cited on pages 16, 95).

[102] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H S Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, 2016 (cited on pages 16, 36).

[103] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019 (cited on pages 17, 36).

[104] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017 (cited on pages 18, 20, 37, 40, 43, 44, 46, 67, 98, 109, 110).

[105] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017 (cited on pages 18, 37, 38).

[106] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014 (cited on pages 19, 102, 109, 130).

[107] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016 (cited on pages 19, 67, 76, 77, 109).

[108] Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016 (cited on pages 19, 67, 76, 77).

[109] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016 (cited on pages 19, 26, 67, 76, 77, 130).

[110] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013 (cited on pages 19, 20).

[111] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *EMNLP*, 2014 (cited on pages 19, 20).

[112] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016 (cited on pages 20, 40, 43).

[113] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv:1804.02767*, 2018 (cited on pages 20, 67).

[114] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020 (cited on pages 20, 128).

[115] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020 (cited on pages 20, 128).

[116] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020 (cited on page 20).

[117] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*, 2016 (cited on pages 20, 109).

[118] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018 (cited on page 21).

[119] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019 (cited on page 21).

[120] `https://github.com/oxford-cs-deepnlp-2017/lectures`. Online; accessed September 2022 (cited on page 21).

[121] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *CVPR*, 2018 (cited on page 21).

[122] Raymond Mooney. *The 'cramming' quote*. `https://www.cs.utexas.edu/~mooney/cramming.html`. Online; accessed September 2022. 2014 (cited on page 22).

[123] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015 (cited on pages 22, 23).

[124] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018 (cited on pages 23, 29, 32, 35, 39, 41).

[125] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015 (cited on page 23).

[126] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016 (cited on page 23).

[127] 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 *CVPR*, 2020 (cited on pages 23, 67, 77, 79, 80).

[128] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder fusion network with co-attention embedding for referring image segmentation. In *CVPR*, 2021 (cited on pages 23, 24, 29, 68, 71, 77, 78, 81).

[129] 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 *CVPR*, 2018 (cited on pages 24, 68, 77, 95).

[130] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *ICCV*, 2019 (cited on pages 24, 25).

[131] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *ICCV*, 2021 (cited on pages 24, 25, 68).

[132] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *ICCV*, 2019 (cited on pages 25, 26).

[133] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016 (cited on page 26).

[134] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016 (cited on page 26).

[135] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. In *IJCV*, 2013 (cited on page 26).

[136] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014 (cited on page 26).

[137] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 2017 (cited on page 26).

[138] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. TransVG: End-to-end visual grounding with Transformers. In *ICCV*, 2021 (cited on page 26).

[139] Chenliang Xu, Shao-Hang Hsieh, Caiming Xiong, and Jason J Corso. Can humans fly? Action understanding with multiple classes of actors. In *CVPR*, 2015 (cited on pages 26, 94, 100, 110, 111).

[140] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV*, 2018 (cited on pages 27, 28, 92, 96, 100–102, 108, 110–112).

[141] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016 (cited on pages 27, 86, 92, 98, 109).

[142] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019 (cited on pages 27, 96).

[143] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *CVPR*, 2019 (cited on pages 27, 95).

[144] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In *ICCV*, 2017 (cited on page 28).

[145] Benjamin Klein, Lior Wolf, and Yehuda Afek. A dynamic convolutional layer for short range weather prediction. In *CVPR*, 2015 (cited on page 28).

[146] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse R-CNN: End-to-end object detection with learnable proposals. In *CVPR*, 2021 (cited on pages 28, 55).

[147] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *ICCV*, 2021 (cited on pages 28, 55).

[148] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. In *TPAMI*, 2021 (cited on pages 29, 71, 77, 96, 103).

[149] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017 (cited on page 29).

[150] Bruce McIntosh, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Visual-textual capsule routing for text-based video segmentation. In *CVPR*, 2020 (cited on pages 29, 96, 101, 103, 111).

[151] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *ICANN*, 2011 (cited on pages 29, 30).

[152] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *ICLR*, 2018 (cited on page 30).

[153] Tianrui Hui, Shaofei Huang, Si Liu, Zihan Ding, Guanbin Li, Wenguan Wang, Jizhong Han, and Fei Wang. Collaborative spatial-temporal modeling for language-queried video actor segmentation. In *CVPR*, 2021 (cited on pages 30, 96, 103).

[154] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, 2019 (cited on page 30).

[155] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022 (cited on pages 30, 31).

[156] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with Transformers. In *CVPR*, 2021 (cited on page 30).

[157] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019 (cited on pages 39, 41).

[158] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *ICCV*, 2015 (cited on page 41).

[159] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019 (cited on page 44).

[160] Qibin Hou, Mingming Cheng, Xiao-Wei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. In *TPAMI*, 2019 (cited on pages 45, 50).

[161] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017 (cited on page 48).

[162] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013 (cited on pages 48, 57).

[163] Dong Lao and Ganesh Sundaramoorthi. Extending layered models to 3D motion. In *ECCV*, 2018 (cited on pages 48, 57).

[164] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *CVPR*, 2017 (cited on page 50).

[165] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *CVPR*, 2017 (cited on page 50).

[166] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *CVPR*, 2017 (cited on page 50).

[167] Liang Lin Guanbin Li Yuan Xie and Yizhou Yu. Instance-level salient object segmentation. In *CVPR*, 2017 (cited on page 50).

[168] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *CVPR*, 2017 (cited on page 50).

[169] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016 (cited on page 50).

[170] Nian Liu and Junwei Han. DHSNet: Deep hierarchical saliency network for salient object detection. In *CVPR*, 2016 (cited on page 50).

[171] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, 2016 (cited on page 50).

[172] Tiantian Wang, Lihe Zhang, Huchuan Lu, Chong Sun, and Jinqing Qi. Kernelized subspace ranking for saliency detection. In *ECCV*, 2016 (cited on page 50).

[173] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016 (cited on page 50).

[174] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow guided recurrent neural encoder for video salient object detection. In *CVPR*, 2018 (cited on page 50).

[175] Wenguan Wang, Jianbing Shen, and Ling Shao. Video salient object detection via fully convolutional networks. In *TIP*, 2018 (cited on page 50).

[176] Zhi Liu, Junhao Li, Linwei Ye, Guangling Sun, and Liquan Shen. Saliency detection for unconstrained videos using superpixel-level graph and spatiotemporal propagation. In *TCSVT*, 2017 (cited on page 50).

[177] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015 (cited on page 50).

[178] Yuming Fang, Zhou Wang, Weisi Lin, and Zhijun Fang. Video saliency incorporating spatiotemporal cues and uncertainty weighting. In *TIP*, 2014 (cited on page 50).

[179] Zhi Liu, Xiang Zhang, Shuhua Luo, and Olivier Le Meur. Superpixel-based spatiotemporal saliency detection. In *TCSVT*, 2014 (cited on page 50).

[180] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020 (cited on pages 54, 131).

[181] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an Atkinson-Shiffrin memory model. In *ECCV*, 2022 (cited on page 54).

[182] Brent A. Griffin and Jason J. Corso. BubbleNets: Learning to select the guidance frame in video object segmentation by deep sorting frames. In *CVPR*, 2019 (cited on page 54).

[183] Zhaoyuan Yin, Jia Zheng, Weixin Luo, Shenhan Qian, Hanling Zhang, and Shenghua Gao. Learning to recommend frame for interactive video object segmentation in the wild. In *CVPR*, 2021 (cited on page 54).

[184] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin Transformer. In *CVPR*, 2022 (cited on pages 55, 68).

[185] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016 (cited on page 56).

[186] Brent A. Griffin and Jason J. Corso. Tukey-inspired video object segmentation. In *WACV*, 2019 (cited on page 57).

[187] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015 (cited on page 57).

[188] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014 (cited on page 57).

[189] Peter Ochs and Thomas Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011 (cited on page 57).

[190] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, 2011 (cited on page 57).

[191] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, 2015 (cited on page 57).

[192] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012 (cited on page 57).

[193] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *CVPR*, 2019 (cited on page 65).

[194] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *CVPR*, 2018 (cited on page 65).

[195] 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 *ECCV*, 2014 (cited on pages 65, 76, 129).

[196] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. In *IJCV*, 2019 (cited on page 65).

[197] Ruiyu Li, Kai-Can Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *CVPR*, 2018 (cited on pages 65, 67, 71, 77, 95).

[198] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021 (cited on pages 65, 68, 85, 128).

[199] Ronghang Hu and Amanpreet Singh. UniT: Multimodal multitask learning with a unified Transformer. In *ICCV*, 2021 (cited on pages 65, 68).

[200] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019 (cited on pages 65, 68, 128).

[201] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: ClipBERT for video-and-language learning via sparse sampling. In *CVPR*, 2021 (cited on pages 65, 68).

[202] Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu Tan. Locate then segment: A strong pipeline for referring image segmentation. In *CVPR*, 2021 (cited on pages 67, 77, 81).

[203] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image Transformers & distillation through attention. In *ICML*, 2021 (cited on page 68).

[204] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision Transformer. In *ICCV*, 2021 (cited on page 68).

[205] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021 (cited on page 68).

[206] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. DenseCLIP: Language-guided dense prediction with context-aware prompting. In *CVPR*, 2022 (cited on page 68).

[207] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006 (cited on pages 68, 128).

[208] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016 (cited on pages 68, 128).

[209] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020 (cited on page 68).

[210] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016 (cited on page 72).

[211] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *ICML*, 2010 (cited on pages 73–75).

[212] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015 (cited on page 75).

[213] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019 (cited on page 75).

[214] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020 (cited on page 75).

[215] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009 (cited on pages 75, 109).

[216] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019 (cited on page 75).

[217] Yi-Wen Chen, Yi-Hsuan Tsai, Tiantian Wang, Yen-Yu Lin, and Ming-Hsuan Yang. Referring expression object segmentation with caption-aware consistency. In *BMVC*, 2019 (cited on page 77).

[218] Sibei Yang, Meng Xia, Guanbin Li, Hong-Yu Zhou, and Yizhou Yu. Bottom-up shift and reasoning for referring image segmentation. In *CVPR*, 2021 (cited on page 77).

[219] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022 (cited on pages 85, 130).

[220] Jaimeen Ahn and Alice Oh. Mitigating language-dependent ethnic bias in BERT. In *EMNLP*, 2021 (cited on page 86).

[221] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017 (cited on pages 86, 94).

[222] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018 (cited on page 86).

[223] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020 (cited on page 86).

[224] Dimitris Chatzopoulos, Carlos Bermejo, Zhanpeng Huang, and Pan Hui. Mobile augmented reality survey: From where we are to where we go. In *IEEE Access*, 2017 (cited on page 92).

[225] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015 (cited on page 92).

[226] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014 (cited on page 92).

[227] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshops*, 2014 (cited on page 92).

[228] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017 (cited on page 94).

[229] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J. Black. Towards understanding action recognition. In *ICCV*, 2013 (cited on pages 94, 100, 110, 111).

[230] 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 *ECCV*, 2020 (cited on page 95).

[231] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020 (cited on pages 96, 110).

[232] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. FEELVOS: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019 (cited on pages 96, 110).

[233] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. UnOVOST: unsupervised offline video object segmentation and tracking. In *WACV*, 2020 (cited on page 96).

[234] Zhao Yang, Qiang Wang, Song Bai, Weiming Hu, and Philip HS Torr. Video segmentation by detection for the 2019 unsupervised DAVIS challenge. In *CVPR Workshops*, 2019 (cited on page 96).

[235] Shubhika Garg and Vidit Goel. Mask selection and propagation for unsupervised video object segmentation. In *WACV*, 2021 (cited on page 96).

[236] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020 (cited on pages 101, 109).

[237] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The Kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017 (cited on pages 103, 110).

[238] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015 (cited on page 109).

[239] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. In *TPAMI*, 2015 (cited on page 109).

[240] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from Transformers. In *EMNLP*, 2019 (cited on page 128).

[241] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-ViL: Knowledge enhanced vision-language representations through scene graph. In *AAAI*, 2021 (cited on page 128).

[242] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal image-text representation learning. In *ECCV*, 2020 (cited on page 128).

[243] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. In *NeurIPS*, 2020 (cited on page 128).

[244] Lyndon Chan, Mahdi S Hosseini, and Konstantinos N Plataniotis. A comprehensive analysis of weakly-supervised semantic segmentation in different image domains. In *IJCV*, 2021 (cited on page 128).

[245] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017 (cited on page 129).

[246] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021 (cited on page 130).

[247] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022 (cited on page 130).

[248] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022 (cited on page 130).

[249] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016 (cited on page 130).

[250] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018 (cited on page 130).

[251] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018 (cited on page 130).

[252] Emiel Krahmer and Kees Van Deemter. Computational generation of referring expressions: A survey. In *Computational Linguistics*, 2012 (cited on page 130).

[253] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016 (cited on page 131).

[254] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. SegSort: Segmentation by discriminative sorting of segments. In *ICCV*, 2019 (cited on page 131).

[255] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019 (cited on page 131).

[256] Yassine Ouali, Céline Hudelot, and Myriam Tami. Autoregressive unsupervised image segmentation. In *ECCV*, 2020 (cited on page 131).

[257] Andrey Voynov, Stanislav Morozov, and Artem Babenko. Object segmentation without labels with large-scale generative models. In *ICML*, 2021 (cited on page 131).

[258] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *ICCV*, 2021 (cited on page 131).

[259] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018 (cited on page 131).

[260] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019 (cited on page 131).

[261] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*, 2019 (cited on page 131).

[262] Zihang Lai, Erika Lu, and Weidi Xie. MAST: A memory-augmented self-supervised tracker. In *CVPR*, 2020 (cited on page 131).

[263] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *ICLR*, 2020 (cited on page 131).

[264] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *CVPR*, 2021 (cited on page 131).

[265] Guillaume Lample and Devendra Singh Chaplot. Playing FPS games with deep reinforcement learning. In *AAAI*, 2017 (cited on page 132).

[266] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. In *Nature*, 2015 (cited on page 132).

[267] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016 (cited on page 132).