Target Localization and Tracking under Multiple Modalities

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Mingjie Sun

June 2022

Abstract

Recent years have seen the remarkable success of Computer Vision (CV) and Natural Language Processing (NLP). This thesis aims at a new task, multi-modal target localization and tracking, closely linking to both aforementioned topics, where the target is to localize or track the object in an image or video when the object information is provided in advance. Such a task attracts increasing attention because it can be widely used in many real-world scenarios, including theft prevention, remote video monitoring, vandalism deterrence, vehicle counting, parking lot surveillance and so on. However, this task is still far from being well tackled, especially in some challenging situations of deformation, motion blur, illumination variation, background transformation, occlusion and so on. Before the localization and tracking process, the target information can be provided under different modalities, including the mask-level modality (*i.e.*, pixel-level segmentation of the target), box-level modality (*i.e.*, minimum enclosing rectangle of the target within the image), text-level modality (*i.e.*, referring sentence to describe the attribution of the target), *etc.*

Therefore, to better analyze and handle this task, it is divided into several sub-tasks in this thesis, including text-referred target localization, text-referred target tracking, box-referred target tracking and mask-referred target tracking, and the merit and shortcoming of each modality are analyzed in this thesis. Mask-level modality can provide the most precise visual information of the target, while it takes too much time to generate a pixel-level segmentation annotation, which greatly restricts its practicability. Box-level modality accelerates the annotation procedure, as it is much easier to draw a box of the target in an image compared with the segmentation mask. However, much noise is introduced in the box annotation, as some non-target regions are also included in the bounding box, leading to inferior performance. Text-level modality can provide the linguistic information for the target, which is highly summarized and abstracted. Even though some details of the target may be ignored, the text annotation is compatible for many frames in a video, while the mask and box annotations do not have such advantage.

In this thesis, based on these advantages and disadvantages of all modalities, new frameworks are designed for each aforementioned sub-task. When dealing with the static image, it is normally required to localize the target according to a descriptive sentence. In terms of the text-referred target localization under a fully-supervised setting, the target localization procedure is viewed as a sequence of iterative shrinking steps, where a non-target region is cut off within each iteration. The remaining region, after several shrinking iterations, is ultimately viewed as the localization result of the target. As the ground-truth label for the optimal action within each iteration is

unknown for the model, reinforcement learning is adopted to train such a model, as it only requires the feedback of the conducted action, instead of the precise action label. When such a task is under a weakly-supervised setting, a discriminative triad based matching and reconstruction are proposed, where a referring sentence to describe the target is parsed into 3-unit triads, with its first, second, third unit indicating the target object, reference object, and their discriminative relation, respectively. Thus, the model can focus on the significant information in the referring sentence, with meaningless phrases ignored. Furthermore, the corresponding triad-based matching and reconstruction mechanism is also designed for inference and optimization for the proposed framework.

When handing the video tracking and segmentation, the target can be depicted under text, box or mask modality. For text-referred target tracking, a start-frame selecting model is designed, as the referring sentence for the target is normally compatible with all frames in a video. In addition, we proposed a multiple standards matching mechanism, including the motion-language based, subject-language based, appearance-based and IoU-based standards, to comprehensively consider all metrics during the matching process. Ultimately, four scores for the aforementioned standards are aggregated into a final matching score, and the candidate proposal with the highest final score is selected as the box result for the target. When it comes to box-referred target tracking, two "smart" switches are proposed. The first one is to control whether to adopt the complex appearance-based matching or simple IoU-based matching. The second one is to control whether to update the target template using the result of the current frame. In this way, a better balance between accuracy and speed is achieved in the proposed framework. Such switches are also implemented using the reinforcement learning agent, as it can consider not only the current profit but also the potential profit in the future, due to the adopted accumulative future reward for training. For mask-referred target tracking, based on previous works using the segmentation result to update the model to gradually adapt to the appearance change of the target, we propose to mine optimal adaptation areas for such online adaptation, and the optimal areas are selected through a reinforcement learning agent trained through the actor-critic framework.

In this way, all sub-tasked of multi-modal target localization and tracking are well tackled in this thesis, and all proposed frameworks can serve as the basis for future works aiming at these tasks.

Key Words: Multiple Modalities, Target Localization, Tracking and Segmentation, Reinforcement Learning

Acknowledgements

My research and education would not have been possible without the help and support of supervisors, colleagues, and family. First and foremost, I am highly grateful for the advice and support from my main supervisors, Professor Eng Gee Lim and Professor Jimin Xiao, who have helped guide me on both academic and personal decisions in the past four years. The meetings and conversations are vital in inspiring me to think outside the box, from multiple perspectives to form a comprehensive and objective critique. Their academic rigour and innovative thinking inspire me along the way of my research. Their guidance has helped me during this research and in the writing of this thesis.

Furthermore, I want to thank my co-supervisor, Professor John Y. Goulermas for his patience and support in supervising me in my research, especially in the profession of writing, whose influence can be seen throughout this thesis.

In addition, I would like to thank my IPAP advisors, Professor Ming Xu for providing guidance and feedback throughout my four-year research period. I am also thankful to my examiners Professor Ming-Ming Cheng and Professor Qiufeng Wang, not only for their insightful comments and encouragements, but also for their tough questions, which force me to widen my research from various perspectives.

Besides my advisor, I extend my thanks to all my friends in Lab 408 and 410: Dr. Yanchun Xie, Mr. Dingyuan Zheng, Mr. Bingfeng Zhang, Miss Hui Li, Miss Siyue Yu, Mr. Xiaoyang Wang, Mr. Xinqiao Zhao, Mr. Jian Wang, Mr. Junkun Peng, Mr. Junwei Wu. I really treasure the days being with them, discussing the ideas, sporting in the gym and dining together after a paper is accepted.

My sincere thanks also go to my family, primarily my parents, who have provided me with the love and support I have needed throughout all of this, and without them, none would have been possible.

Last but not least, I would like to thank my wife, my little piggy, Mengyi Wang, for your everyday good night and morning, for your smiling face when having my dishes, for your caring when I feel sick, for cheering me up when I am in a bad mood, and for being my muse with all your love in my research life. Without your tremendous understanding and encouragement in the past few months, it would be impossible for me to complete this thesis. And most of all, thank you for choosing to be my partner. I owe you everything.

Contents

Abstract **i**

Acknowledgements **iii**

Contents **vii**

List of Figures **xi**

List of Tables **xiv**

List of Acronyms **xvi**

1 Introduction **1**
1.1 Research Motivation 1
1.2 Research Questions 3
1.3 Main Contributions 4
1.4 Thesis Outline 6

2 Literature Review **7**
2.1 Multi-modal Target Localization 7
2.1.1 Two-stage Target Localization 7
2.1.2 One-stage Target Localization 8
2.1.3 Weakly Target Localization 9
2.2 Multi-modal Target Tracking 10
2.2.1 Mask-referred Target Tracking 10
2.2.2 Text-referred Target Tracking 11
2.3 Reinforcement Learning 12

3 Text-referred Target Localization in Supervised Learning **14**
3.1 Motivation 14
3.2 Methodology 17
3.2.1 Overview 17

3.2.2 Agent State . . . 18
3.2.3 Action and Reward . . . 20
3.2.4 Actor-critic Training . . . 22
3.2.5 Implementation Details . . . 23
3.3 Experiments . . . 24
3.3.1 Datasets and Metrics . . . 24
3.3.2 Comparison with State-of-the-art . . . 24
3.3.3 Visualization of Reasoning Process . . . 28
3.3.4 Ablation Studies . . . 28

4 Text-referred Target Localization in Weakly Learning 30
4.1 Discriminative Triad Based Matching and Reconstruction . . . 30
4.1.1 Motivation . . . 30
4.1.2 Methodology . . . 32
4.1.3 Experiments . . . 38
4.2 Cycle-free Mechanism . . . 44
4.2.1 Motivation . . . 44
4.2.2 Methodology . . . 47
4.2.3 Experiments . . . 52

5 Text-referred Target Tracking 60
5.1 Motivation . . . 60
5.2 Methodology . . . 62
5.2.1 Overview . . . 62
5.2.2 Starting Frame Selection . . . 63
5.2.3 Matching Score Calculation . . . 65
5.2.4 Matching Score Aggregation . . . 68
5.3 Implementation Details . . . 69
5.3.1 Starting Frame Selection . . . 69
5.3.2 Matching Scores . . . 70
5.3.3 Segmentation Network . . . 70
5.4 Experiments . . . 70
5.4.1 Experimental Setup . . . 70
5.4.2 Comparison with State-of-the-art . . . 72
5.4.3 Ablation Studies . . . 73

6 Box-referred Target Tracking 78
6.1 Motivation . . . 78
6.2 Methodology . . . 81
6.2.1 Overview . . . 81
6.2.2 Agent Action . . . 81

6.2.3 State and Reward . . . 83
6.2.4 Search Region Size . . . 85
6.2.5 Actor-Critic Training . . . 85
6.3 Implementation Details . . . 87
6.3.1 Segmentation Network Training . . . 87
6.3.2 RL Agent Training . . . 87
6.4 Experiments . . . 88
6.4.1 Experiment Setup . . . 88
6.4.2 Comparison with State-of-the-art . . . 90
6.4.3 Ablation studies . . . 91

7 Mask-referred Target Tracking 94
7.1 Motivation . . . 94
7.2 Methodology . . . 95
7.2.1 Overview . . . 95
7.2.2 Agent Action . . . 99
7.2.3 State and Reward . . . 100
7.2.4 Training in Actor-Critic Framework . . . 102
7.3 Implementation Details . . . 103
7.3.1 Train the Segmentation Network . . . 103
7.3.2 Train the RL Model . . . 103
7.3.3 Accelerating RL Training . . . 103
7.4 Experiments . . . 106
7.4.1 Experiment Setup . . . 106
7.4.2 Comparison with State-of-the-art . . . 106
7.4.3 Ablation studies . . . 108

8 Conclusion 115
8.1 Summary . . . 115
8.2 Future Works . . . 116

Publication List 117

Bibliography 119

List of Figures

1.1 Applications of the automatic target localization and tracking model. 1

1.2 Illustration of the target information under different modalities. 2

1.3 Framework outline for all tasks in this thesis (chapter 3 - 7). 6

3.1 Illustration of the difference between (a) existing feature-point level matching methods, and (b) the proposed iterative shrinking method. The query sentence for the target in this image is "the cat above the shelf". In (a), dots depict the feature points within the image feature map, with the surrounding area representing its corresponding image region. Red dots indicate the feature points whose corresponding image regions may cover the target (*i.e.*, the cat on the shelf) or distracting objects (*i.e.*, cats in other places), while black dots are unrelated feature points. It can be observed that there is no such feature point whose image region can cover both the target object (*i.e.*, cat) and the reference object (*i.e.*, shelf). In (b), for each iteration, the area in the dashed box indicates the image patch after shrinking, and the red arrow depicts the shrinking direction, which is decided by comprehensively considering all contents in the image patch before shrinking. . . 16

3.2 Architecture of our method where the blue lines indicate the states corresponding to different scale levels, and the yellow circle represents the optimizing of the actor and critic networks within each iteration. 17

3.3 Illustration of our action design. $\{a^1, a^2, a^3, a^4\}$ corresponds to image shrinking toward 4 different directions, respectively. a^5 corresponds to the STOP action. . . 20

3.4 Visualization of the reasoning process about how to iteratively localize the target. The image patch generated through shrinking is marked in red box. The parts above the images indicate the triads parsed from the query, with the black units representing the contents remaining in the corresponding image patch, and the grey units indicating that the visual contents of the corresponding distracting or reference objects are already removed. 26

3.5 Some failure cases of the proposed method where the green box is the ground-truth one, and the red box is the predicted one which is incorrect. 26

4.1 Illustration of the difference between (a) the traditional weakly text-referred target localization method, and (b) the proposed triad-level method. The blue and yellow parts indicate the proposal features and the linguistic feature, while the red represents the individual loss function. In (a), although the meaning of the reconstructed sentence is the same as the label, the loss is large, as it is calculated word-by-word, making the network hard to converge. In (b) however, the reconstruction is conducted at the triad level, and a 300-D MSE loss function replaces the 10,000-D CE loss function which dramatically facilitates the weakly-supervised training. 32

4.2 The proposed framework mainly consists of the generation of discriminative triads, the triad matching module and the triad reconstruction module (three rightmost modules in the bottom row). Other modules are also illustrated, including the analysis of the sentence structure (leftmost module in the bottom row) and the organization of proposal pairs (modules in the top row). The parts in solid red, green and blue boxes represent the features of a proposal pair used to match the linguistics feature of the target unit, the discriminative unit and the reference unit in a discriminative triad, respectively. Note that the parts in dashed boxes, whose specific routes are omitted, work in the same way as the parts in solid boxes. . . . 34

4.3 Visualization results on MSCOCO datasets, where the red, green, blue parts represent the target, discriminative, reference unit and their corresponding proposals, respectively. 40

4.4 The qualitative comparison with other weakly text-referred target localization methods.The red box indicates the bounding box result of the target. 42

4.5 The diagram illustrates the difference between the traditional cyclic forward-backward weakly text-referred target localization pipeline and the proposed cycle-free weakly text-referred target localization pipeline: (a) is the given image and query sentence with the target region localized by the green dashed box, (b) is the traditional forward-back pipeline and (c) is the proposed cycle-free pipeline, where “diff” indicates the difference between the query sentence and the textual description. Note that (c) only provides an overview idea of the proposed method, rather than its detailed implementation. Best viewed in colors. 45

4.6 The architecture diagram of the proposed method. The upper part is related to the visual information processing, including the proposal generation from the given image using a detection network, the formation of the proposal pairs using the candidate proposals, and the region describer network to generate the textual description of each proposal pair. The bottom left part is about the textual information processing, including the sentence parsing to generate the word dependency and the generation of the discriminative triads. The bottom right part corresponds to the optimization strategy of the proposed method, including the minimum distance filter to select the positive sample with the minimum distance, and the self-paced learning strategy to judge if the filtered sample is suitable to optimize the network. Best viewed in colors. 46
4.7 The qualitative comparison results between the proposed method and other weakly text-referred target localization methods, where green boxes indicate the labels and red boxes indicate the predictions. Best viewed in colors. 55
4.8 Some visualization results of the proposed method: (a) targets correctly localized by the proposed method where the prediction is in the red box, (b) some failed cases of the proposed method where the prediction is in the red box and the ground truth is in the green box. Best viewed in colors. 56
4.9 Examples of the deviation issue. Column (a) consists of the image, query, target (localized in green box) and a triad unit. The intermediate results of DTMR [99] and the proposed method are illustrated in (b) and (c), respectively. The value in blue corresponds to the region or region pair in blue box, the same as the yellow one. The "score" in (b) is the matching score calculated by the forward module, while "diff" in (c) is the distance between the description and query. Note that unrelated regions are ignored. 57

5.1 A segmentation label is exclusive to only one frame, while a language label is typically compatible with all frames in a video, allowing to select a more suitable frame as the starting point. This video is from the DAVIS 2017 dataset [85], and only one frame is used for language annotation. 61
5.2 Illustration of: (a) the architecture of S-Net ($n = 8$); (b) the way in which n frames are selected in round 1 and round 2, where grids in black or gray indicate the selected frames as the input for S-Net, and grids in black indicate the frame with the highest score predicted by S-Net ($n = 8$). 62
5.3 Illustration of the segmentation part of the proposed method. First, the motion-language score, subject-language score, appearance score and IOU score for each candidate proposal are calculated through the M-net, visual grounding network, A-net and IOU values. Then, these four scores are aggregated using weights generated by W-net into a final matching score in an attention-based way to determine the most possible proposal corresponding to the target. Foreground segmentation is conducted on this proposal to predict the final target mask (best viewed in color). 63

5.4 Architectural illustration of (a) motion matching network (M-Net), (b) appearance matching network (A-Net), (c) weight generating network (W-Net). 67
5.5 In this diagram, the red box in the current frame indicates the correct location of the target, while the green box in the previous frame indicates the predicted target location. ML, SL, IOU, APPE and MEAN are the motion-language, subject-language-matching, IOU, appearance-matching and final average score, respectively. "weight" is the obtained weight value for each standard generated through W-Net. Instead of showing all candidate proposals, only major proposals in a frame are shown in this diagram. 74

6.1 Speed-accuracy trade-off for various target tracking and segmentation methods on DAVIS 2017 dataset. Methods in green only rely on the first-frame box-level ground-truth while methods in red rely on the first-frame pixel-level ground-truth. 79
6.2 Architecture of our method where unconditional paths (full line) indicate they will be conducted in any situation, while conditional paths (dashed line) indicates they will be conducted only in some particular situations. 80
6.3 Illustration of the minimum enclosing rectangle b_m and search region b_s. b_s is generated by b_m by expansion. 83
6.4 Illustration of elements in the target template (T_{box}, T_{mask}, $T_{box'}$, $T_{mask'}$). Elements in the predicted result (P_{box}, P_{mask}, $P_{box'}$, $P_{mask'}$) are in the same form. 85
6.5 The RL state consists of the features of S_T and S_P. 86
6.6 Speed-accuracy trade-off on the VOT 2018 Long-term dataset [47]. 92

7.1 Different adaptation areas lead to different frame segmentation results. The segmentation models are the same before online adaptation. In the left column, the area in green is the adaptation area for foreground and the area in purple is the adaptation area for background. 96
7.2 The network architecture of our work consists of two RL models. One is to choose the adaptation area for the foreground, and another is to choose the adaptation area for the background. We use pre-trained VGG19 model to extract the feature of each ROI area of the current frame, and then feed the combined feature into the RL model as the state. Finally, the RL model will choose the best adaptation area and update the segmentation model using the chosen adaptation area. 97
7.3 Foreground adaptation ROI selecting action set A^p and background adaptation ROI selecting action set A^n. 100
7.4 The data structure used to restore the related data to accelerate the RL model training. For RL model to select the t_n, distance map is required for the training, while the training for the RL mode to selected t_p does not need it. 105
7.5 Visualization of segmentation masks and the mean region similarity J_m for different methods on the DAVIS 2016 dataset. 109

List of Tables

3.1 Common queries and their corresponding discriminative triads on the RefCOCOg, RefCOCO+ and RefCOCO datasets. **ID** is the index of a discriminative triad for a query, as a complex query may generate multiple discriminative triads. **T/R/D** indicates the target, reference, and discriminative unit, respectively. 19

3.2 Performance comparison between our method and other text-referred target localization methods on the RefCLEF dataset. The method with the highest accuracy is in bold. 23

3.3 Accuracy comparison against other text-referred target localization methods. "PF" indicates proposal-free text-referred target localization methods. Specifically, methods on the top half of the table are not proposal-free, while methods on the bottom half are proposal-free. "Ave" is the average accuracy of RefCOCOg, RefCOCO+, and RefCOCO, where the mean accuracy of testA and testB is used for RefCOCO+ and RefCOCO. The method with the highest accuracy is in bold. . 25

3.4 Ablation studies on RefCOCO (testA) and RefCOCOg (test) datasets. The method with the highest accuracy is in bold. 27

4.1 Effectiveness comparison with other weakly text-referred target localization methods. The method with the highest score is in bold. "det" indicates that the candidate proposals generated from Faster RCNN [88]. In other cases, the candidate proposals are provided by the bounding box ground truth. 39

4.2 Ablation studies through cross-valuation. 43

4.3 Performance comparison with other weakly text-referred target localization methods. "GT" indicates that the candidate proposals are from the bounding box ground truth. In other cases, the candidate proposals are generated from Faster RCNN [88]. "Ave" is the average accuracy of RefCOCO and RefCOCO+ datasets, where the mean accuracy of val, testA and testB sets for each dataset is calculated firstly, through which the final average accuracy for RefCOCO and RefCOCO+ datasets is then obtained. The best score is highlighted in bold, and the second-best score is marked with underline. 53

4.4 Ablation studies through cross validation on the RefCOCO dataset. The setting with the best score is highlighted in bold. 54

5.1 Quantitative comparison with other related methods on DAVIS 2017 (Da17), DAVIS 2016 (Da16), SegTrack V2 (ST) and Youtube-Object (YO) datasets using the metrics of mean region similarity (J) and the average score of region similarity and boundary similarity ($J\&F$). "Expr" is the task that uses the first-frame language annotation for initialization, and "Mask" is the task that uses the first-frame segmentation annotation for initialization. 71

5.2 Ablation study of starting frame on DAVIS 2017. 73

5.3 Ablation study of matching standards on DAVIS 2017. ML, SL, APPE and IOU indicate the motion-language, subject-language, appearance matching and IOU score, respectively. 75

5.4 Ablation study on the contribution of motion information on DAVIS 2017 with our new annotations. 75

6.1 Quantitative comparison with other methods on the DAVIS 2017 (D7), DAVIS 2016 (D6), SegTrack V2 (S) and Youtube-Object (Y) datasets, measured by the mean region similarity (J), as well as the average score of region similarity and boundary similarity ($J\&F$). **FT** indicates fine-tuning, **M** indicates using the first-frame masks, t(s) indicates the average running time per frame in seconds. The method with the best score is bold, and the method with the second best score is marked in underline. 89

6.2 Ablation studies on the DAVIS 2017 dataset, measured by the mean region similarity (J_m). 91

7.1 Quantitative comparison with other methods on the DAVIS 2016, SegTrack V2 and Youtube-Object dataset. For J_m and F_m, the method wiht the best performance is bold, and the method with the second best performance is marked with underline. 104

7.2 Ablation study on the contribution of individual RL model for the DAVIS 2016 dataset, measured by the mean region similarity J_m. WO indicates without. 107

7.3 Ablation study on the contribution of individual RL model for the SegTrack V2 dataset, measured by the mean region similarity J_m. WO indicates without. 108

7.4 Performance comparison between heuristic adaptation ROIs selection and RL-based adaptation ROIs selection, conducted on the DAVIS 2016 dataset, measured by the mean region similarity J_m. t_n indicates the threshold to choose the adaptation ROI for background. t_p indicates the threshold to choose the adaptation ROI for foreground. 111

7.5 The quality comparison between the selected ROIs of the proposed method and the baseline method, as well as their influences on the final segmentation results, conducted on the DAVIS 2016 dataset, measured by the mean region similarity J_m. In this table, IOU indicates the IOU between the adaptation ROI selected and the ground-truth. J_m indicates the mean region similarity. B refers to the background and F refers to the foreground. RL refers to the RL model and Base refers to the baseline method. Δ refers to the difference value between the RL model and the baseline method. All values in this table are calculated before CRF. 113

List of Acronyms

AC Actor Critic

BP Back Propagation

CNN Convolutional Neural Network

CV Computer Vision

IoU Intersection over Union

LSTM Long short-Term Memory

MDP Markov Decision Process

NLP Natural Language Processing

NN Normal Noun

NP Noun Phrase

PG Policy Gradient

RL Reinforcement Learning

RNN Recursive Neural Network

RoI Region of Interest

RPN Region Proposal Network

SOTA State-of-the-art

Chapter 1

Introduction

1.1 Research Motivation

Target surveillance, aiming at achieving the real-time observation of the person or vehicle within a busy environment, has emerged as an important research task in the computer vision community. Given the target information, target surveillance consists of two key processes. One process is to localize the target, which focuses on comprehending the target information to distinguish the target from other distracting objects. Another process is to track the target, which concentrates on propagating the target information from the current frame to the next ones. Thus, such a task is also named automatic target localization and tracking.

Figure 1.1: Applications of the automatic target localization and tracking model.

The growth of such a task is being driven by a sharp rise in the expenditure of human resources to find out the target in the imageries or videos captured by the mounting cameras. In the past, although such cameras are already equipped in most public places (*e.g.*, banks, stores, and parking lots), the captured imagery or videos used to be utilized "after the fact". As for an example, after a

crime occurs, say a car is stolen or a store is robbed, the police can only go back to these captured imageries or videos, recorded in tapes, to see what happened several hours ago, which completely loses its primary benefit as an active, real-time medium. However, nowadays there is a better choice to train a model to automatically localize and track the suspicious targets loitering in the store or parking lot, when options are still open for avoiding the potential crime.

Apart from the police to search the suspect, as can be observed in Fig. 1.1, the target localization and tracking model can also be utilized in many common facilities, including monitoring pedestrian congestion, detecting car accidents, measuring traffic flow on highways, and counting endangered species in forests. Such a model can be adopted for military purposes as well, including monitoring peace treaties, patrolling national borders, providing secure perimeters around bases and embassies and measuring the flow of refugees.

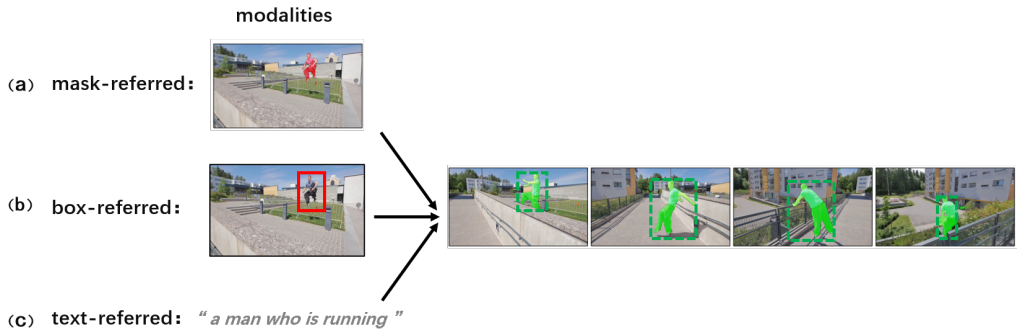

Figure 1.2: Illustration of the target information under different modalities.

To smoothly localize and track the target, its information is required to be provided to the model under a certain modality in advance. Common modalities to depict the target include the mask, box and text annotations, *etc.* As can be seen from Fig. 1.2.(a), mask annotation indicates the pixel-level segmentation label for the target. As the annotation for the target under this modality is exactly correct and without any noise introduced, existing methods relying on this modality to provide the target information usually perform better when compared with methods adopting the target information under other modalities. However, such methods tend to be impractical when dealing with real-world scenarios, as generating the pixel-level segmentation annotation of the target is too time-consuming. It normally takes one or two minutes to annotate a person or vehicle. Such time cost is greatly broadened when there are complicated contours of the target. Thus, it tends formidable for these methods to be applied in reality when a large number of targets need to be annotated at the pixel level within a limited time.

To overcome this shortcoming, a substitute modality is the box annotation. As shown in Fig. 1.2.(b), such a modality indicates the minimum enclosing rectangle of the target in the image. Obviously, compared with the mask annotation, the time cost for annotating the target can be greatly reduced when adopting the box-level modality, as it only takes one or two seconds to draw a bounding box of the target in the image. Whereas, methods adopting the box-level modality suffer from the noise introduced by the bounding box annotation, as there are also some non-target regions exist in the bounding box, leading to a requirement for the model to distinguish the target from the background region by itself when comprehending and remembering the target information, which greatly limits the performance of these methods.

Apart from the aforementioned modalities, the text annotation is another common modality, illustrated in Fig. 1.2.(c). Normally, the text annotation is a referring sentence to describe the attributions of the target. Similar to box annotation, text annotation can also be generated within a few seconds. In addition, the unique merit of this modality is that, different from the mask or box annotation which is exclusive to a certain frame in the video, the descriptive sentence is usually compatible with many frames in the video, because the textual description is extremely highly-summarised information for the target. Also, it is easier for the textual sentence to describe the motion information of the target (*e.g.*, move left), compared with other modalities. As the target information is highly abstracted in the referring sentence, there is no doubt that some details are ignored in this modality. When the lost information serves as the key to distinguishing the target from other distracting objects, localization or tracking failures go common for these methods.

Therefore, in this thesis, a number of frameworks are designed to localize or track the target under different modalities to make the best use of the merit of each modality and avoid their disadvantages as much as possible.

1.2 Research Questions

In this thesis, a number of research questions about multiple modalities are tackled, and different research questions are analyzed in different chapters, as listed:

In chapter 3, the fully-supervised text-referred target localization task is carefully studied [98], where a referring sentence is utilized to localize the target in an image, and the mapping between box labels and sentence labels is available during training. Existing methods tend to employ a

query-image matching branch to select the highest-score point in the image feature map as the target box centre, with its width and height predicted by another branch. Then, the main research question to be investigated is how to avoid the disadvantage of previous methods that they fail to utilize the contextual relation between the target and reference objects, and lack interpretability in their reasoning procedure.

In chapter 4, we further explore the weakly supervised text-referred target localization task [99, 102], where the mapping between box labels and sentences is not available during training. In traditional methods, an object region that best matches the referring expression is picked out, and then the query sentence is reconstructed from the selected region, where the reconstruction difference serves as the loss for back-propagation. The main research question to be investigated in this chapter is how to guarantee the correctness of the matching process, serving as the key for the next reconstruction process.

In chapter 5, the text-referred target tracking task is studied [103], where the first-frame language annotation is used to describe the target object in the video. Firstly, we investigate the feasibility to choose the most suitable starting frame to mitigate initialization failure, as a language label is typically compatible with all frames in a video. Then, the complementarity of different matching standards is further explored.

In chapter 6, we study the box-referred target tracking task [101], where the first-frame bounding box label of the target is provided. The main research question to be investigated in this chapter is how to make the optimal selection of the matching method to predict the result as well as to decide whether to update the target template using the newly predicted result.

In chapter 7, the task of mask-referred target tracking is explored [100], where the first-frame segmentation annotation of the target is provided. In this task, the main research question lies in how to improve the rough and inflexible model updating mechanism of previous methods.

1.3 Main Contributions

The main contributions of this thesis are analyzing the properties and drawbacks of different modalities for the target localization and tracking model, and new frameworks are designed for each sub-task to take full advantage of the merit of each modality.

Fully-supervised Text-referred Target Localization: We formalize the localization process as a sequence of image-level shrinking processes, with the shrinking direction in each iteration decided by a reinforcement leaning agent after comprehensively considering all objects in the image patch, allowing the relation information in a query to be fully utilized. More details can be seen in [98].

Weakly-supervised Text-referred Target Localization: Firstly, we propose a discriminative triad and a scalable query parsing strategy, where a referring sentence can be converted into one or multiple discriminative triads with the same formulation, with each triad instructing how to recognize a target object from a reference object. Based on the discriminative triad, the triad-level matching and reconstruction modules are specifically designed. Secondly, we also propose a cycle-free framework to tackle this task, where a region describer is designed to predict the textual description for each candidate region, and a result region is selected according to the linguistic feature distance between each description and the initial query sentence. In turn, the region describer network is optimized using the result region with the minimum feature distance. More details can be seen in [99, 102].

Text-referred Target Tracking: We first propose a mechanism to select the most suitable starting frame rather than the first frame as the starting point for each video. Then, we introduce a new standard for the matching process: the similarity score between a moving target in the video and a language label with motion information. To calculate this score, the optical flow is extracted to represent the motion information in a video, and scores from different standards are aggregated using a flexible and attention-based method. More details can be seen in [103].

Box-referred Target Tracking: We first provide a simple way to trade off between running speed and accuracy by selecting the matching method (IOU-based matching or appearance-based matching). Then, we adopt a reinforcement learning agent to make the decision on whether to update the target template or not according to the quality of the predicted result, which effectively prevents the drift problem and boots the accuracy of our method. More details can be seen in [101].

Mask-referred Target Tracking: We propose to mine optimal adaptation areas for online adaptation. Specifically, we design a reinforcement learning based framework to train the agent for generating the adaptation areas. Furthermore, to speed up the training of the reinforcement learning network, we also design a novel multi-branch tree-based exploration method to fast select the best state action pairs. More details can be seen in [100].

Chapter	Task	Target Information	Input	Learning Setting
3	text-referred target localization	sentence *" a man who is running "*	image	fully-supervised
4	text-referred target localization	sentence *" a man who is running "*	image	weakly-supervised
5	text-referred target tracking	sentence *" a man who is running "*	video	fully-supervised
6	box-referred target tracking	bounding box	video	fully-supervised
7	mask-referred target tracking	segmentation mask	video	fully-supervised

Figure 1.3: Framework outline for all tasks in this thesis (chapter 3 - 7).

1.4 Thesis Outline

The remainder of the thesis is structured as follows. A literature review is conducted in Chapter 2, and the proposed framework for each sub-task is described in Chapters 3, 4, 5, 6 and 7, as can be seen in Fig. 1.3. Specifically, chapter 3 introduces an iterative shrinking mechanism to tackle the fully-supervised text-referred target localization task, where the image iteratively shrinks into the correct target region. Chapter 4 mainly describes discriminative triad representation for the weakly-supervised text-referred target localization task, through which a query sentence can be parsed into one or multiple discriminative triads in a very scalable way. Chapter 5 introduces a starting frame selection mechanism for the text-referred target tracking task to take full advantage of the characteristic that language labels are typically compatible with all frames in a video. Chapter 6 introduces a novel framework that can select the matching method and target template update mechanism for each frame to improve the performance of the box-referred target tracking task. Chapter 7 describes a mechanism that can select the best adaptation region of interest for each frame to take full advantage of the test time adaptation for the mask-referred target tracking task. Chapter 8 gathers the conclusions of this thesis and gives recommendations for future research.

Chapter 2

Literature Review

In this chapter, a literature review about target localization and tracking methods is conducted. Existing works of each sub-task are described in detail, including multi-modal target localization and multi-modal target tracking. In addition, as RL is frequently used to tackle these tasks, some related works are also discussed in this chapter.

2.1 Multi-modal Target Localization

Text-referred target localization is the task to recognize and localize a target within an image described by a query sentence (*i.e.*, referring expression), where the mapping between proposals and queries is available during the training stage, requiring the composite knowledge of both computer vision and natural language process domain. Existing works can be divided into two groups, including two-stage and one-stage methods. In the two-stage setting, candidate proposals are beforehand generated through a detection network (*e.g.,* Faster RCNN [88]), where additional datasets are required to pre-train the detector. While in the one-stage setting, the bounding box of the target is predicted by the text-referred target localization network itself, without any additional dataset for pre-training.

2.1.1 Two-stage Target Localization

Two-stage methods [2, 127, 141, 1] firstly extract candidate proposals in the image, and then rank them according to their similarities with the query sentence. Then, the proposal with the highest similarity score is selected as the final prediction. Traditionally, the entire query sentence is directly encoded through a single language embedding network (*e.g.*, LSTM[34]), without considering the

variance among different types of query sentences [90, 114, 73, 133]. To avoid treating a query sentence as a single unit, [132] proposes an attention mechanism to decompose the query sentence into three linguistic modular components, related to subject appearance, location and relationship to other objects. Then, three corresponding visual modules are adopted to generate the subject appearance feature, location feature and relationship feature for each candidate proposal in the image. Finally, three matching scores are calculated to select the best-matched proposal as the final prediction. To prevent the attention mechanism from focusing on only the most dominant features of both modalities, and ignoring other potential linguistic-visual correspondences, an erasing model is proposed in [66], where the most dominant linguistic or visual information is discarded to drive the model to discover more complementary linguistic-visual correspondences. To better exploit the relationship information between the target object and its neighborhoods, [116] proposes a graph-based network, where the graph nodes correspond to object text-referred target localization in the image, and graph edges represent relationships between these objects. In this way, the representation of each object proposal can be enriched with the additional relationship information of neighborhoods, and better adapt to the linguistic query sentence. Similarly, the gated graph neural network [54] is adopted in [126] to better compute the semantic context of every proposal and the global context of the query sentence.

2.1.2 One-stage Target Localization

As the performance of two-stage methods is heavily restricted by the quality of the candidate proposals generated in the first stage, some works attempt to adopt a one-stage strategy without the stage to generate the candidate proposals before the matching process, which significantly accelerates the inference speed. [92] combines the linguistic feature of the query sentence and the visual feature from the image to generate a multi-modal feature map, and predict the binary classification scores (foreground or background) and regression parameters. Following [61], 9 anchors are adopted for every index of the multi-model feature map, and the anchor with the highest score is chosen as the final prediction after regression. Similarly, [129] modifies another one-stage object detector, YOLOV3 [87], to fuse the linguistic feature from the query sentence to the visual feature of the image. Apart from the linguistic and visual features, the spatial feature is also utilized to account for spatial mentions in query sentences. Finally, the sigmod output layer in [87] is replaced by a softmax function to generate only one image region in response to a query. [58] adopts a correlation filtering module to calculate the similarity with the query sentence for every index in the visual feature of the image, and predict the centre of the target according

to the similarity score. Then, the size and offset (distance difference between high-solution and high-solution coordinates) are predicted by another network to generate the final bounding box of the target.

Although fully-supervised methods perform well on existing datasets, annotating the connection of query-proposal pairs is too time-consuming to tackle real-life issues. Thus, more and more researchers attempt to handle this task in a weakly-supervised way.

2.1.3 Weakly Target Localization

Different from the supervised text-referred target localization task, weakly text-referred target localization only requires image-level correspondences. In other words, there is no mapping between image regions and the query sentence during the training stage.

To tackle this task, the early work [90] proposes to predict the attention scores between the query sentence and all candidate proposals. Then, reconstruct the query sentence using the weighted sum over the visual features of all proposals using their attention weights. The difference between the original query sentence and the reconstructed one is utilized for training the model. In the test time, the image proposal with the highest attention score is selected as the prediction.

Based on this framework, apart from the reconstruction loss from the language modality, the consistency contained in the visual modality is also utilized in [10]. In other words, the location parameters of each candidate proposal are reconstructed, and the difference between the original location and the reconstructed one is utilized for training as well.

Inspired by the related fully-supervised text-referred target localization method [132], which separates the query feature into the subject, location and context components, Liu *et al.* [67] attempts to optimize the network by minimizing the distance between the linguistic feature of each component and its corresponding visual feature, apart from the reconstruction loss from the entire query sentence using the framework in [90]. As the separation of the query is inherited directly from the full supervised text-referred target localization method [132], this mechanism performs not that well in the weakly-supervised setting. Also, the separation strategy is designed artificially, not learned by the model itself during training, heavily restricting the universality of the network. To simplify the separation mechanism to adapt to the weakly-supervised setting, [68] parses each query into a subject-object pair, and the matching process is conducted between the linguistic subject-object pair and visual proposal-proposal pair. Note that the category information of each proposal is utilized to remove some unrelated proposals before the matching process. Although it proposes a simple matching model, which seems easy to train in the weakly-supervised

setting, it cannot tackle some complicated query sentences which cannot be simply parsed into a subject-object pair. Also, the whole network is still optimized relying on reconstructing the entire sentence, using a heavy RNN-style network that occupies too many parameters of the entire network.

2.2 Multi-modal Target Tracking

2.2.1 Mask-referred Target Tracking

Mask-referred target tracking aims at generating the segmentation mask of the target in each frame of a video sequence when the pixel-level target annotation in the first frame is available. Early methods focus on the single-target video sequence where only one target appears in the video. A convolution network, pre-trained on ImageNet [23] dataset, is adopted in [7] to generate a general segmentation model using the target tracking and segmentation datasets [83, 85]. When encountering a certain video sequence, the first-frame pixel-level annotation of the target is used to fine-tune the segmentation model so that it can be aware of the target information and predict the segmentation masks of the remaining frames. To make the segmentation model gradually adjust to the appearance change of the target in the video, an online adaptation mechanism is proposed in [111], where some high-confidence regions, both for foreground and background, are used to update the segmentation after the inference of a certain frame. Thus, the segmentation model can adapt to the target appearance variation when deformation occurs or the camera rotates. Following such a pipeline, a dynamic update mechanism is further proposed in [57], where new features are absorbed and obsolete features are discarded through an adaptive feature bank. In [55], a cyclic loss and gradient correction module are proposed to mitigate error propagation, with the result of the performance and speed trade-off.

Recently, the multi-target scenarios, where several targets are required to be tracked and segmented simultaneously, attracts increasing attention. Early methods tend to handle each target one by one and consume multiple times computing resources. AOT [130] makes the first attempt to propose a simple yet effective mechanism to associate, match, and decode all targets in the video simultaneously, where the pixel-level templates of all targets are firstly embedded into the same high-dimensional embedding space for further propagation through the identification embedding mechanism. Then, the aggregated feature is passed into the segmentation network and all targets' probabilities map can be decoded from the network's output feature. Thus, AOT can simultaneously generate all target results through one inference process.

STM [79] proposes the first space-time memory network, where previous frames and their corresponding mask results (image-mask pairs) are firstly sorted in an external memory, and then the segmentation result is predicted according to the affinity between the current frame and the previous image-mask pairs stored in the memory. As the segmentation mask of each target, as well as its corresponding frame image, are separately encoded through a heavy memory encoder, STM [79] also suffers from the increasing computing cost when dealing with the multi-target scenarios. Therefore, STCN [16] modifies STM [79] and proposes to firstly establish correspondences directly between the current frame and the previous frames (direct image-to-image affinities). And then the masks in the memory are encoded using a light decoder. Ultimately, the segmentation result of a certain target is predicted according to the image-to-image affinity and the previous frames' masks in the memory. In addition, instead of the inner-product affinity which leads to poor use of memory, STCN [16] proposes to use the negative squared Euclidean distance to compute the affinity.

Another improvement based on STM [79] is HMMN [95] which proposes two advanced memory read modules. Firstly, different from existing memory-based methods where the matching process is conducted only on the highest layer (*e.g.*, 1/16 resolution) output, the matching process in HMMN [95] is conducted in multiple layers. Secondly, a hierarchical memory matching scheme is proposed where the memory reading on a low-layer (*e.g.*, 1/8 resolution) is guided by that on a high-layer (*e.g.*, 1/16 resolution). JOINT [74] adopts a unified framework to combine transductive reasoning and online inductive learning together. In transductive reasoning, the pixel-level affinity or distance map between the current frame and previous frames is obtained to conduct the direct reasoning from previous mask results to the current one. While online inductive learning utilizes the reference frame and its corresponding annotated mask to train a segmentation model with the target information involved, which is then applied to subsequent frames to predict the segmentation mask.

2.2.2 Text-referred Target Tracking

Different form mask-referred target tracking where the first-frame annotation is provided in the pixel-level, text-referred target tracking only provides a referring sentence to describe the target in the video. Early text-referred target tracking methods rely on the off-the-shelf instance segmentation network to predict the result. The language annotations of [83, 85] are firstly provided in [43]. In this work, the referring sentence and frame image are firstly passed into a pre-trained visual grounding network [139, 132], with the results of a number of candidate proposals

with their corresponding matching scores, indicating whether a certain proposal corresponds to the referring sentence. However, such matching scores are only conducted at the static image level, rather than the video level. Thus, each box proposal is re-ranked based on its overlap with the bounding box result in the previous frame to improve temporal consistency and reduce id-switches for target object predictions across the whole video.

Instead of proposal-sentence matching, [56] proposes a tracklet-sentence matching mechanism and achieves better performance. In detail, a set of candidate proposals of several key frames, sampled over the entire video sequence, are firstly generated by the pre-trained instance segmentation network. Then, these candidate proposals are propagated to the whole video and a set of object tracklets are generated. After that, a tracklet-NMS mechanism is designed to remove redundant tracklets and select the high-quality ones as candidates for language-guided segmentation. In the tracklet-sentence matching stage, a transformer-based grounding module is designed and the tracklet with the highest score is selected as the result.

The off-the-shelf instance segmentation network is discarded in [5], where the visual feature of the frame image is extracted through DeepLabv3 [11] with ResNet101 [33] backbone, and the linguistic feature of the referring sentence is extracted through Bert [24]. However, such a method treats each frame in the video as an individual image, without any temporal consistency considered.

To solve such an issue, YOFO [50] proposes the process all frames in a video simultaneously with the temporal consistency utilized. In detail, after the visual and linguistic features are generated through the extractors, cross-modal feature mining module is proposed to generate the bi-modal feature and soft weight map. Then, its proposed Meta-Transfer module is learned to transfer the target-specific information from the bi-modal feature to the image modality, and the transferred image feature is ultimately sent to the decoder for the final segmentation prediction.

2.3 Reinforcement Learning

RL aims at training an agent to evaluate the impact of a certain action under a particular state, and it is effective to optimize sequential decision problems. In other words, RL algorithm learns to achieve a complex objective from past experience. "actor-critic" [46] is a popular RL framework that inherits several previous RL frameworks including deep Q-learning [78] and policy gradient [105] which are valued-based and policy-based strategies, respectively.

RL has been applied to many areas of computer vision, in particular for visual object tracking at the bounding box level. In [135], Yun *et al.*. use RL to choose sequential actions to move

the bounding box step by step from the original object location in the previous frame to the correct location. In [37], Huang *et al.* propose a novel RL based model which can choose the most suitable number of deep convolutional layers according to the current frame complexity, which dramatically reduces the running time. In [25], Dong *et al.* propose to deploy the RL to choose the optimal hyper-parameter for correlation filter.

In terms of multi-modal tasks, Rennie *et al.* [89] applies the policy gradient to train the image caption network based on its baseline method [124]. Specifically, the image caption network is viewed as an RL agent, whose action is to predict the next word. The state of the RL agent is the cells and hidden variables of the LSTM, as well as their corresponding attention weights, etc. The reward function is designed according to the sentence-level evaluation metrics, CIDEr score [109], representing the difference between the generated and ground-truth sentence. Yu *et al.* [134] adopts RL to guide the referring sentence generator towards less ambiguous expressions. Different from [89], the reward function is designed according to the similarity score between the generated sentence and the object image patch, calculated by a pre-trained region-language matching network. However, RL is never tried to tackle the investigated text-referred target localization task, as all existing text-referred target localization methods follow the static region-sentence matching strategy, leaving no space to make any discrete decision. In this chapter, we discard the traditional matching strategy, and make the first attempt to formulate text-referred target localization as a sequential shrinking process, where an RL agent is trained to predict the shrinking direction within each iteration.

Chapter 3

Text-referred Target Localization in Supervised Learning

In this chapter, a new framework is designed for text-referred target localization, aiming at recognize and localizing the target object in an image according to its query sentence (referring expression), which requires joint comprehension in both visual and linguistic domains. text-referred target localization is a fundamental multi-modality task, serving as the basis for many downstream tasks, including the visual question answering [40, 12, 44, 119, 49, 3, 106], image caption [28, 13, 19, 80, 140, 107, 93], image-text matching [63, 140, 9, 138, 14, 113, 52], etc. It attracts much attention from both computer vision (CV) and natural language processing (NLP) areas in recent years.

3.1 Motivation

Conventional methods formulate text-referred target localization as a region-retrieval problem [132, 134, 126, 116, 64, 67, 29], with proposals of all candidate objects provided in advance. The candidate proposals can be either provided from the bounding box ground-truth or generated by a pre-trained object detector (*e.g.*, Faster RCNN [88]). The matching networks of these text-referred target localization models predict the similarity score between the query and each proposal, with the highest-score proposal selected as the target object. In this way, conventional text-referred target localization methods highly rely on the ground-truth bounding boxes or an accurate object detector which requires a lot of extra data to train it in advance. To overcome these shortcomings, some proposal-free methods attempt to predict the text-referred target localization result without

object proposals.

All existing proposal-free text-referred target localization methods [58, 129, 92, 128] directly follow the pipeline of the one-stage detector (*e.g.*, YOLOv3 [87]) and adopt a two-branch style network, with the first branch to calculate the similarity score between the query sentence and each feature point within the image feature map, and the second branch to generate its bounding box coordinates. The predicted bounding box of the feature point with the highest score is viewed as the final result.

These methods, with the matching process conducted between the query sentence and each image feature point, perform well for simple queries with the target described by its own attribute (*e.g.*, "man in blue"), but it is difficult for them to deal with complex queries, especially when the target is referred by another reference object. As can be observed from Fig.3.1.(a), since there is no such a feature point whose corresponding image region covers both the target "cat" and the reference object "shelf", the contextual relation between the target and the reference object, serving as the key to distinguish the target from other distracting objects, cannot be fully utilized in such a matching process. Thus, the text-referred target localization performance is greatly compromised.

Another shortcoming of these methods is the lack of interpretability. As the matching network predicts the final matching scores of all feature points in one step, its inner reasoning procedure is hidden and invisible. Thus, if the matching model fails, it is formidable to analyze the causes.

To tackle these problems simultaneously, we formalize text-referred target localization as a sequence of image-level shrinking processes. Within each iteration, the image shrinks along a certain direction, with a non-target image region removed. After several shrinking iterations, only the target image region remains, and it is viewed as the final result. The shrinking direction in each iteration is predicted by a trainable network. As the optimal shrinking direction is uncertain for each iteration, conventional supervised training is not suited. We propose to model it as a Markov decision process [38] and adopt reinforcement learning (RL) to tackle it. RL only requires a "feedback" or "reward" after each shrinking step, rather than the exact supervision label. Besides, RL considers not only the current reward but also the potential reward in future, which further improves the performance.

Our proposed method can better utilize the contextual relation between the target and reference objects because the shrinking direction is decided by comprehensively considering all objects within the current image patch. For an instance, as shown in Fig.3.1.(b), in the first iteration, the image region where a black cat lies on the chair is removed (*i.e.*, shrinking towards the right), because "chair" does not occur in the query. In the second iteration, after analyzing the spatial

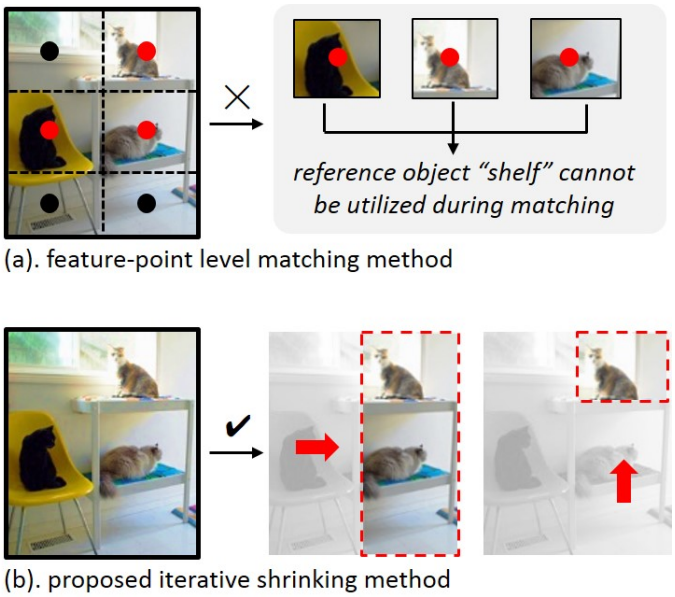

Figure 3.1: Illustration of the difference between (a) existing feature-point level matching methods, and (b) the proposed iterative shrinking method. The query sentence for the target in this image is "the cat above the shelf". In (a), dots depict the feature points within the image feature map, with the surrounding area representing its corresponding image region. Red dots indicate the feature points whose corresponding image regions may cover the target (*i.e.*, the cat on the shelf) or distracting objects (*i.e.*, cats in other places), while black dots are unrelated feature points. It can be observed that there is no such feature point whose image region can cover both the target object (*i.e.*, cat) and the reference object (*i.e.*, shelf). In (b), for each iteration, the area in the dashed box indicates the image patch after shrinking, and the red arrow depicts the shrinking direction, which is decided by comprehensively considering all contents in the image patch before shrinking.

relationship between each candidate cat and the reference object "shelf", it is decided to remove the image region where a cat lies inside the shelf (*i.e.*, shrinking towards the top), due to its mismatching spatial relation against the query. Ultimately, only the image region accurately covering the target cat remains. As the relation information in a query is fully utilized, the proposed method can better deal with complex queries and images (*e.g.*, RefCOCOg [73]), whilst existing matching methods fail to handle such cases (Fig.3.1.(a)).

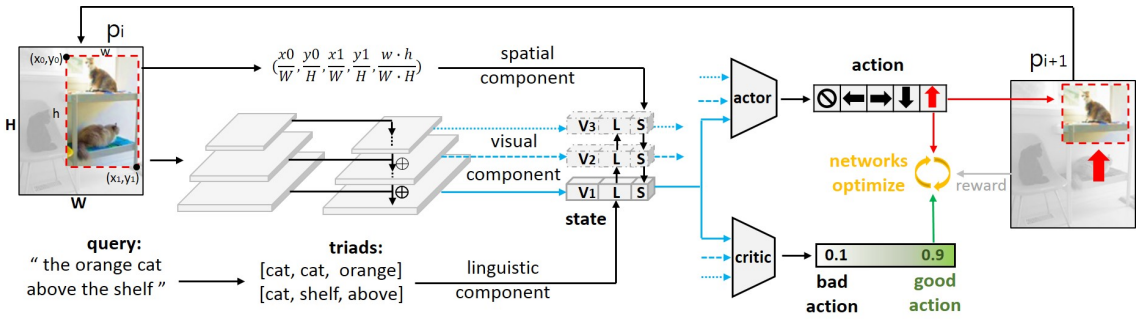

Figure 3.2: Architecture of our method where the blue lines indicate the states corresponding to different scale levels, and the yellow circle represents the optimizing of the actor and critic networks within each iteration.

3.2 Methodology

3.2.1 Overview

Given the image *img* and the query sentence q, the target object is localized through a sequence of shrinking iterations, with i indicating the index of each iteration. Specifically, for a certain iteration i, the current image patch p_i shrinks along a certain direction, decided by the RL agent, into a smaller patch p_{i+1}, with a non-target region eliminated. After several sequential iterations, all non-target regions are removed and only the region exactly covering the target object remains, which is viewed as the final text-referred target localization result. Note that $p_1 = img$ at the beginning.

In terms of the shrinking direction selection, we adopt an RL agent to make the decision within each iteration, and view the aforementioned shrinking procedure as a Markov decision process (MDP) [38]. Specifically, within a certain iteration i, the state s_i is generated according to the information embedded in p_i and q (Section 3.2.2). Then, according to s_i, the RL agent predicts an action a_i (shrinking direction or stop). After a_i is executed, a reward r_i is calculated as a "feedback" of a_i (Section 3.2.3). Ultimately, the RL agent is trained through an actor-critic RL framework, where the RL agent consists of two components, including an actor network to predict the action, and a critic network to evaluate the predicted action (Section 3.2.4). The architecture of our proposed method is shown in Fig.3.2.

3.2.2 Agent State

In conventional MDP, states indicate the observations that an agent receives from the external environment. Specifically to our method, for a certain image patch p_i within iteration i, the state s_i is the input passed into the RL agent to predict the action, consisting of three components, including the linguistic feature f_q^l extracted from q, visual feature $f_{p_i}^v$ extracted from p_i, and spatial feature $f_{p_i}^s$ extracted from the relative position of p_i within *img*.

Linguistic Component

To extract the linguistic feature f_q^l, q is first parsed into multiple discriminative triads $\{t_k\}_{k=1}^M$ [99]. Each triad represents a piece of discriminative information to distinguish the target from the distracting or reference objects. As can be observed from Table 3.1, a discriminative triad t_k consists of three components, including a target unit t_k^t, a reference unit t_k^r and a discriminative unit t_k^d, denoting the target object, the reference object against the target object and the discriminative relation between them, respectively. Note that, the discriminative unit t_k^d is defined in a broad sense to take full advantage of the potential discriminative information hidden in queries. As for an instance, when it comes to "orange cat" in Table 3.1, this query implies that there are cats in other colors with a high probability. Thus, "cat" is set as its reference unit to represent the distracting cats in other colors.

For its implementation, an off-the-shelf NLP processing toolbox (*e.g.*, Stanford CoreNLP [8] or Spacy [35]) is employed to analyze the tree structure of each query, as well as the dependency label of each word in the query. In general, triads are generated within two steps. The first step is to extract the rightmost Normal Noun (NN) of the bottom-left Noun Phrase (NP) in the sentence tree as the target unit shared by all triads. The second step is to generate the reference and discriminative units using different parsing patterns. As for an instance, the subject-relation-object phrase, such as "the man holding a cat", firstly the second component of a particular dependency, whose first component is the same as the target unit, and the dependency type is nsubj, is viewed as a discriminative unit. Then, the second component of another dependency, whose first component is the same as that of the current discriminative unit, is viewed as its corresponding reference unit.

Then, a pre-trained word vector generation method (*e.g.*, Word2vec [76] or Glove [81]) is adopted to extract the linguistic embedding $f_{t_k^t}^l$, $f_{t_k^r}^l$, $f_{t_k^d}^l$, corresponding to t_k^t, t_k^r, t_k^d, respectively. The linguistic feature $f_{t_k}^l$ of a certain triad t_k is the concatenation of its three components' linguistic features:

$$f_{t_k}^l = f_{t_k^t}^l \oplus f_{t_k^r}^l \oplus f_{t_k^d}^l, \tag{3.1}$$

Table 3.1: Common queries and their corresponding discriminative triads on the RefCOCOg, RefCOCO+ and RefCOCO datasets. **ID** is the index of a discriminative triad for a query, as a complex query may generate multiple discriminative triads. **T/R/D** indicates the target, reference, and discriminative unit, respectively.

Query	**ID**	**T**	**R**	**D**
"lady"	1	lady	lady	SELF
"left"	1	UKN	UKN	left
"left lady"	1	lady	lady	left
"orange cat"	1	cat	cat	orange
"cat above a shelf"	1	cat	shelf	above
"lady holding a cat"	1	lady	cat	holding
"the left lady in white holding an orange cat and standing on a table"	1	lady	lady	left
	2	lady	lady	white
	3	lady	table	on
	4	lady	cat	holding
	5	cat	cat	orange

where $\oplus$ indicates the concatenation operation. Similarly, another concatenation operation is further conducted on all triads' linguistic features:

$$f_q^l = f_{t_1}^l \oplus f_{t_2}^l \oplus \cdots \oplus f_{t_M}^l, \tag{3.2}$$

and f_q^l represents the ultimate linguistic feature of q. Note that, if the total number of the triads parsed from a query is less than M, the first triad feature $f_{t_1}^l$ is repeated until the triad number reaches M.

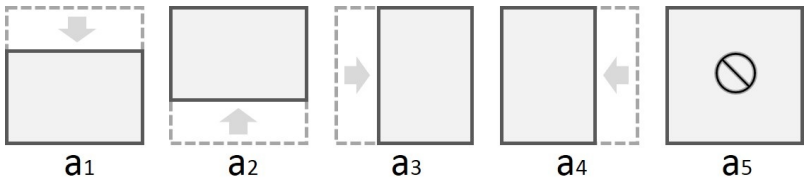

Figure 3.3: Illustration of our action design. $\{a^1, a^2, a^3, a^4\}$ corresponds to image shrinking toward 4 different directions, respectively. a^5 corresponds to the STOP action.

Visual and Spatial Components

The visual feature of the image patch p_i is extracted through an off-the-shell convolutional network (*e.g.*, VGG [96] or ResNet [33]) and the Feature Pyramid Network (FPN) [59]. Specifically, the convolutional network, *e.g.*, ResNet [33], first generates three raw visual feature maps, $f_{p_i}^{\tilde{v}_1}$, $f_{p_i}^{\tilde{v}_2}$, $f_{p_i}^{\tilde{v}_3}$, with different scales of $[28 \times 28 \times 512]$, $[14 \times 14 \times 1024]$ and $[7 \times 7 \times 2048]$, respectively. Then, the up-sampling, fusing and average-pooling operations of FPN [59] are conducted on these three raw feature maps, which generates three one-hot vectors with the same dimension (D_v), $f_{p_i}^{v_1}$, $f_{p_i}^{v_2}$ and $f_{p_i}^{v_2}$, as the corresponding visual feature for each scale level.

The spatial feature $f_{p_i}^s$ of the image patch p_i is encoded by a 5-D vector, $[\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{w \cdot h}{W \cdot H}]$, where $[x_{tl}, y_{tl}]$ and $[x_{br}, y_{br}]$ refer to the top-left and bottom-right coordinates of p_i within *img*, respectively. w and h indicate the width and height of p_i, while W and H denote the width and height of *img*. $f_{p_i}^s$ can provide the relative position of p_i within *img*, through which the RL agent can infer the relative location of the target object within *img*.

The ultimate state s_i consists of three sub-states s_i^c with $c \in \{1, 2, 3\}$, each of which corresponds to a certain scale level c. Each sub-state is designed as the concatenation of the corresponding linguistic, visual and spatial features:

$$s_i^c = f_q^l \oplus f_{p_i}^{v_c} \oplus f_{p_i}^s. \tag{3.3}$$

3.2.3 Action and Reward

In MDP, after observing the environment, an agent takes action to interact with the external environment. A reward is given to provide the feedback of the conducted action. As shown in Fig.3.3, within a certain iteration i, the RL agent selects an action a_i among the action pool A, consisting of 5 candidate actions, $\{a^1, a^2, a^3, a^4, a^5\}$. The first 4 candidate actions, $\{a^1, a^2, a^3, a^4\}$,

indicate the top side of p_i shrinks towards down, the bottom side of p_i shrinks towards up, the left side of p_i shrinks towards the right and the right side of p_i shrinks towards left, respectively. a_5 indicates the end of the shrinking process (*i.e.*, $p_i = p_{i+1}$), and the current image patch is viewed as the final text-referred target localization result.

To naturally introduce the shrinking stride decay in an episode, the value of the shrinking stride is decided according to the width and height of the image patch p_i before shrinking:

$$\begin{aligned}
\hat{y}_{tl} &= y_{tl} + \alpha(y_{br} - y_{tl}), \text{ if } a_i = a^1 \\
\hat{y}_{br} &= y_{br} - \alpha(y_{br} - y_{tl}), \text{ if } a_i = a^2 \\
\hat{x}_{tl} &= x_{tl} + \alpha(x_{br} - x_{tl}), \text{ if } a_i = a^3 \\
\hat{x}_{br} &= x_{br} - \alpha(x_{br} - x_{tl}), \text{ if } a_i = a^4,
\end{aligned} \tag{3.4}$$

where $[x_{tl}, y_{tl}]$ and $[x_{br}, y_{br}]$ indicate the coordinates of the top-left and bottom-right points of p_i within *img*. $[\hat{x}_{tl}, \hat{y}_{tl}]$ and $[\hat{x}_{br}, \hat{y}_{br}]$ indicate the coordinates of the top-left and bottom-right points of p_{i+1} within in *img*. In this way, at the early stage of an episode, relatively large shrinking strides are adopted to quickly "move" towards to the target. While in the late stage, relatively small shrinking strides are adopted to meticulously localize the accurate position of the target.

To predict such an action a_i, the state s_i^c for each scale level is passed into the RL agent respectively, generating three action probability vectors, with the average one serving as the final probability vector. The action with the highest probability is selected as a_i.

The reward r_i is designed according to the overlap between the image patch after shrinking and the ground-truth bounding box, as follows:

$$r_i = \begin{cases} 0 & IoU < 0.3 \ \ or \ \ \Delta_{IoU} \leqslant 0 \\ 1 & 0.3 \leqslant IoU < 0.5 \ \ and \ \ 0 < \Delta_{IoU} \\ 10 & 0.5 \leqslant IoU \ \ and \ \ 0 < \Delta_{IoU} \end{cases}, \tag{3.5}$$

where IoU refers to the intersection-over-union between the image region after shrinking and the ground-truth bounding box of the target, and Δ_{IoU} indicates the IoU difference between the current iteration and the previous iteration. We set a major difference between the good and bad action's rewards, so as to facilitate the training of the RL agent.

3.2.4 Actor-critic Training

The proposed RL agent is designed and optimized under the "actor-critic" framework [46], consisting of two networks, including the actor network to predict the action among the candidate action pool, and the critic network to predict the score of a certain state, which is further utilized to evaluate the quality of the predicted action. During each iteration of training, the actor and critic networks are optimized in order, while in the inference stage, only the actor network is used to predict the action to localize the target.

Specifically, the actor network is first optimized in a policy-based way:

$$\theta = \theta' + l_a \nabla (log \pi_{\theta'}(s_i, a_i)) A(s_i, a_i), \tag{3.6}$$

where

$$A(s_i, a_i) = \delta_i = r_i + \gamma V_{w'}(s_{i+1}) - V_{w'}(s_i). \tag{3.7}$$

In Eq.(3.6) and Eq.(3.7), θ' and θ indicate the weights of the actor network before and after the update. l_a is the learning rate corresponding to the actor network. $\pi(s, a)$ is the actor network's policy function to predict the probability of taking action a for state s. $A(s, a)$ is the advantage function, which is equal to the TD error function in the "actor-critic" framework [77]. r_i is the reward calculated using Eq.(3.5), and γ refers to the discount factor. w' is the weight of the critic network before the update of the critic network. $V(s)$ indicates the predicted score for state s, generated by the critic network.

After the actor network is updated, the critic network is optimized in a value-based way:

$$w = w' + l_c \delta_i \nabla_{w'} V_{w'}(s_i). \tag{3.8}$$

In Eq.(3.8), w' and w indicate the weights of the critic network before and after the optimization. l_c is the learning rate corresponding to the critic network. δ_i is the TD error function calculated in Eq.(3.7).

In this way, the joint optimization of the actor and critic networks can avoid the disadvantages of both value-based and policy-based methods during training. In other words, the proposed RL agent can be optimized after each iteration, rather than waiting until the end of each episode, which dramatically speeds up the training process yet maintains training stability.

Table 3.2: Performance comparison between our method and other text-referred target localization methods on the RefCLEF dataset. The method with the highest accuracy is in bold.

Method	Accuracy
CITE[84]$_{\text{ECCV18}}$	34.13
IGOP[131]$_{\text{NIPS17}}$	34.70
RCCF[58]$_{\text{CVPR20}}$	63.79
Ours	**65.48**

3.2.5 Implementation Details

To generate the state, Stanford CoreNLP [8] is adopted to parse the query sentence into triads, and Glove word vector [81] is adopted to generate the linguistic feature of each triad. The triad number M for each query is set as 2, because a too small M cannot contain enough information from queries, while an extremely large M may greatly increase the network complexity. Resnet101 [33] is used to generate the raw visual feature of the image patch, and FPN [59] is adopted to generate the fused visual feature, with D_v set as 512. In terms of the action, α is set as 0.2 to generate a moderate shrinking stride. During the training, γ is set as 0.9, and the learning rate is set as 1e-7 for both the actor and critic networks. ADAM [45] is adopted as the optimizer. Training is conducted on one Titan RTX GPU for around 800,000 episodes.

Similar as [58], to obtain a more precise bounding box for a target, we employ a post-processing box refinement model after the shrinking process, where the input is *img* and the predicted bounding box from the shrinking process, and the output is the new refined bounding box. Such box refinement network is trained using the L2 loss between the coordinates of a ground-truth box in MS-COCO [62] and the coordinates of the corresponding sampled box obtained by adding Gaussian noise to it.

3.3 Experiments

3.3.1 Datasets and Metrics

The proposed method is evaluated on four common datasets including RefCOCO [133], RefCOCO+ [133], RefCOCOg [73] and RefCLEF [41]. The first three datasets are collected from MS-COCO [62]. The main difference between RefCOCO+ and RefCOCO is that the former one forbids the absolute location words when generating the query sentences, requiring more attention to the appearance discrimination. RefCOCOg contains longer expressions for both appearance and locations. Referring sentences in RefCLEF are annotated casually, so it is normally used as an auxiliary dataset. The metric used to evaluate the accuracy of the proposed method is similar to the object detection task, where the intersection-over-union (IoU) between the predicted bounding box and the ground-truth one is calculated. Then, the one with IoU higher than 0.5 is treated as correct.

3.3.2 Comparison with State-of-the-art

To evaluate our proposed method, we compare it with other SOTA text-referred target localization methods, which are divided into two groups. The first group, shown in the top half of Table 3.3, consists of text-referred target localization methods with object proposals, including SLR [134], DGA [127] and MAttNet [132]. The second group, shown in the bottom half of Table 3.3, is composed of proposal-free text-referred target localization methods, including FAOAVG [129], MCN [71] and RCCF [58]. As can be observed from Table 3.3, the proposed method achieves a large accuracy gain (4.32%) over the previous SOTA method, RCCF [58], on the test set of the RefCOCOg dataset. When the backbone is replaced by VGG, the proposed method achieves 66.32% on this dataset. The cause of a such big accuracy gain is that, queries in RefCOCOg are much longer and more complex than queries in other datasets, with many targets referred by other neighboring reference objects, namely "targets with reference". Thus, the key to localize such targets is to take full advantage of the relation information between the target and reference objects in queries, which is difficult for the feature-point level matching mechanism [129, 71, 58]. Nevertheless, the image-level shrinking mechanism allows the proposed method to consider all objects within the image patch, and better utilize such contextual relation information, so as to achieve higher accuracy.

Then, similar accuracy improvement can be seen on RefCOCO+ (1.32%) and RefClef (1.69%) datasets, where many queries are complicated as well, as shown in Table 3.3 and Table 3.2. However,

Table 3.3: Accuracy comparison against other text-referred target localization methods. "PF" indicates proposal-free text-referred target localization methods. Specifically, methods on the top half of the table are not proposal-free, while methods on the bottom half are proposal-free. "Ave" is the average accuracy of RefCOCOg, RefCOCO+, and RefCOCO, where the mean accuracy of testA and testB is used for RefCOCO+ and RefCOCO. The method with the highest accuracy is in bold.

Method	PF	Visual Extractor	Ave	RefCOCOg	RefCOCO+		RefCOCO	
				test	testA	testB	testA	testB
SLR[134]	×	ResNet-101	61.25	59.63	60.74	48.80	73.71	64.96
DGA[127]		VGG-16	65.26	63.28	69.07	51.99	78.42	65.53
MAttNet[132]		ResNet-101	68.33	67.01	70.26	56.00	80.43	69.28
FAOAVG[129]	✔	DarkNet-53	61.79	57.45	60.56	52.86	74.81	67.59
MCN[71]		VGG-16	65.69	62.29	65.24	54.26	76.97	**73.09**
RCCF[58]		DLA-34	67.30	65.73	70.35	56.32	**78.93**	66.73
Ours		ResNet-101	**68.63**	**70.05**	**71.05**	**58.25**	74.27	68.10

* RefCOCO results of RCCF are from Table 4, setting 7 in [58], where the feature backbone is obtained without the detection pretraining using the additional classification labels of the 80 categories in MS-COCO dataset. As the results of RefCOCOg and RefCOCO+ are not provided in Table 4, results from Table 3 in [58] are used instead, but the comparison is actually unfavorable for the proposed method. The darknet-based MCN [71] is not included in this table for the same reason.

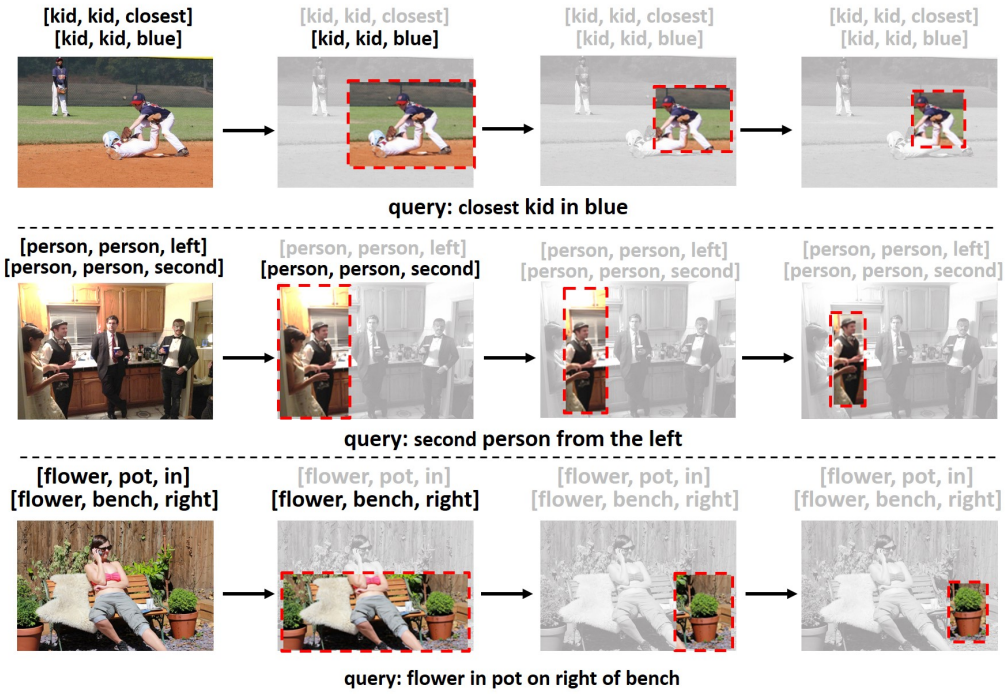

Figure 3.4: Visualization of the reasoning process about how to iteratively localize the target. The image patch generated through shrinking is marked in red box. The parts above the images indicate the triads parsed from the query, with the black units representing the contents remaining in the corresponding image patch, and the grey units indicating that the visual contents of the corresponding distracting or reference objects are already removed.

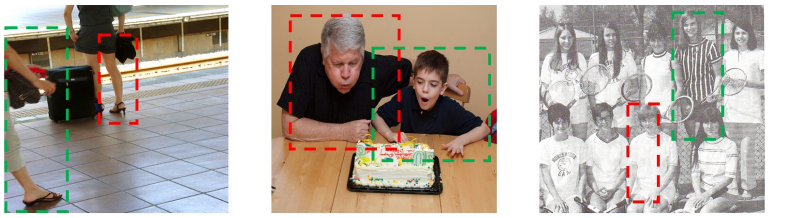

Figure 3.5: Some failure cases of the proposed method where the green box is the ground-truth one, and the red box is the predicted one which is incorrect.

Table 3.4: Ablation studies on RefCOCO (testA) and RefCOCOg (test) datasets. The method with the highest accuracy is in bold.

Setting	RefCOCO	RefCOCOg
w/o triad	67.52	60.17
w/o multi-scale	73.33	66.35
w/o spatial feature	62.42	54.88
fixed stride	45.70	39.46
w/o refinement	70.18	65.62
supervised training	63.84	61.36
Ours	**74.27**	**70.05**

the accuracy gains on RefCOCO+ and RefClef are not as large as that on RefCOCOg. We believe the reason is that the proportion of "targets with reference" in RefCOCO+, as well as RefClef, is relatively smaller than that in RefCOCOg. Ultimately, when it comes to the RefCOCO dataset, where most queries are simple and "targets with reference" are rare, the proposed method only achieves a comparable accuracy, as the superiority of better utilizing the contextual relation information turns useless.

Aforementioned evaluation results on different datasets further prove the positive correlation between the accuracy gain of the proposed method and the number of "targets with reference". Though we have not achieved accuracy gain on all datasets, the average accuracy obtained by the proposed method, on the RefCOCOg, RefCOCO+ and RefCOCO datasets, is still higher than the previous SOTA method RCCF [58] by 1.33%.

Apart from the accuracy gain, it can also be observed from Table 3.3 that the performance of the proposed method is more stable and robust on different datasets compared with other methods. For instance, RCCF [58] achieves quite a high accuracy (78.93%) on testA of the RefCOCO dataset, but it does not perform well on the RefCOCOg dataset (65.73%), with a gap of more than 13%. The proposed method's performance is much more stable, with the corresponding accuracy variation less than 5%, demonstrating the robustness of our proposed method.

3.3.3 Visualization of Reasoning Process

The reasoning process about how to iteratively find and localize the target can be visualized through the sequential shrinking processes, as shown in Fig.3.4. For instance, in the first case in Fig.3.4, the first step of the RL agent is to remove the first distracting object, the kid far from us, which is against the query information "closest", making its corresponding triad [kid,kid,closest] grey. Then, the second step is to eliminate the second distracting object, the kid in white, which is against the query information "blue", making its corresponding triad [kid,kid,blue] grey. Ultimately, after a few small-stride shrinks, the precise target region can be obtained. Note that in the third case, the triad [flower,pot,in] turns grey because the distracting flowers, which are not in the pot, are removed from the image. In general, the distracting and reference objects are iteratively excluded in the early stage, and then the target region is gradually refined, until the exact bounding box of the target is obtained.

Some failure cases of the proposed methods are also reported in Fig.3.5. In Fig.3.5.(a), the heavily-occluded target is difficult to localize. Besides, the tiny facial difference is also hard to be recognized, like the man and the boy wearing the same clothes in Fig.3.5.(b). A similar challenge is also encountered when dealing with the difference in the textual description of the clothing styles, as shown in Fig.3.5.(c). We believe the reason for these failure cases is that similar situations are very rare in the training set, making them quite tough to be handled for text-referred target localization methods. We will try to address these hard cases in our future works.

3.3.4 Ablation Studies

RL State Design

The first ablation study is to explore the contribution of the linguistic, visual and spatial components in the proposed state, respectively. As shown in Table 3.4, for the linguistic component, the triad-based feature is replaced with the feature extracted through Bert [24] (**w/o triad**). Then, the multi-scale feature maps are not utilized, with the raw feature extracted from the CNN network directly employed as the final visual feature (**w/o multi-scale**), to investigate the contribution of the multi-scale visual feature. Lastly, the spatial component is removed from the state to check its influence (**w/o spatial**). The proposed method shows an accuracy gain against all aforementioned settings, which demonstrates the significance of each component in the proposed RL state. Note that the accuracy gain against **w/o triad** comes from better feature representation for the query sentence using the triads, as trivial words in the query (*e.g.*, articles, conjunctions) are

eliminated, and the keywords are organized into a unified format, allowing a better multi-modal comprehension.

RL Action Design

The second ablation study is to study different designs for the agent action. As shown in Table 3.4, firstly, the adaptive shrinking stride is replaced with the fixed shrinking stride (**fixed stride**). Specifically, within each iteration, the shrinking stride is fixed as αH for a_1 and a_2, and αW for a_3 and a_4, where H and W are the width and height of the original image img, respectively, and α is set as 0.2, being the same as in the proposed adaptive shrinking mechanism. The performance drops dramatically in this setting, demonstrating an adaptive shrinking stride contributes significantly to precise localization. Apart from the actions in the shrinking process, we further disable the bounding box refinement network described in Section 3.2.5 (**w/o refinement**), with the comparison result demonstrating the contribution of the bounding box refinement step.

RL or Supervised Learning?

The third ablation study is to explore whether the proposed method can be trained in a supervised way, and we introduce **supervised training** in Table 3.4. In this setting, we randomly sample a region, which covers the whole target object box and its size is larger than the target object box. Then, one side of the sampled region, among the top, bottom, left and right sides, with the highest spatial difference against the target region, is set as the positive label indicating the shrinking direction, if the difference is greater than a threshold. Otherwise, the stop action is set as the positive label. As can be observed in Table 3.4, the proposed RL performs much better than the supervised training. We believe that the major reason is that the adopted sampling strategy for the supervised training cannot cover all situations in reality. In addition, the RL training can consider not only the current profit but also the potential profit in the future, making a higher overall performance.

Chapter 4

Text-referred Target Localization in Weakly Learning

Traditional text-referred target localization methods are trained in a supervised way [132, 126], where the mapping between proposals and queries is available during the training stage. However, drawing the connection between each query sentence and its corresponding proposal is notably time-consuming. Therefore, it has practical significance to handle the text-referred target localization task in a weakly-supervised setting, where the aforementioned mapping is no longer available during training.

4.1 Discriminative Triad Based Matching and Reconstruction

4.1.1 Motivation

Generally, existing weakly text-referred target localization methods consist of two steps; namely, the sentence-level matching and the reconstruction [68, 67, 28, 27]. In the first step, weakly text-referred target localization methods roughly assume the sentence-level matching procedures from existing fully-supervised text-referred target localization methods [132] in order to calculate the similarity between the entire query and each candidate proposal. Specifically, they either adopt an oversimplified sentence-level matching module where a query is roughly parsed into a subject word and an object word [68], making it thus cumbersome to handle complex queries with many descriptive terms, or adopt an over-complicated matching module with multiple language processing sub-networks attached to analyze complex queries [67]. Although these supererogatory

sub-networks barely hinder fully-supervised training, they become problematic under the weakly-supervised setting, as the correctness of the matching result is not guaranteed and this makes the preceding short back-propagation (BP) path unavailable and the new path longer and noisier. Therefore, the simple model tends to perform better than a complicated one [142, 123]. The predicament arising here is that a good weakly text-referred target localization network requires not only a strong and all-purpose matching module to deal with various queries, and especially so with fragmentary and complex ones, but also a delicate and easy-to-train matching module to facilitate the necessitated weakly-supervised training.

The second step is to build a BP path for weakly-supervised training, where a reconstruction stage is created to rebuild the query information with the difference between the original and reconstructed information being the BP loss. Such difference can be calculated either at a sentence level by rebuilding the entire sentence word-by-word, or based on the extracted keyword features. All existing weakly text-referred target localization methods adopt the former one, where a descriptive sentence is predicted from an image region via an RNN-style network, as shown in Fig.4.1.(a). Its accuracy, however, proves hardly satisfactory even for a full-supervised setting [28, 27] and makes the BP loss unreliable. Additionally, we have the architectural imbalance from the heavy RNN-style reconstruction network, which is never used in the final inference stage while occupying a large proportion of parameters of the entire network (around 75% in [68, 67]).

To address these issues simultaneously, we design a discriminative triad as well as a scalable query parsing strategy where a query sentence can be represented by one or multiple discriminative triads with similar formulations. Specifically, the discriminative triad is a 3-unit set, with the first unit representing the target object, the second unit denoting the reference object against the target object, and the third unit indicating the *discriminative relation* between the target object and the reference object. Note that *discriminative relation* is defined in quite a broad sense, including not only the straightforward relation information extracted from a subject-relation-object query, such as “the cat on the table” with the triad {cat, table, on}, but also concealed information hidden in a unary query with descriptive terms, such as “the left man in black”. This query can be parsed to the two triads {man, man, black} and {man, man, left}; see Fig.4.1.(b). In this way, the discriminative triad can represent various forms of queries to facilitate the matching process.

Contrary to the traditional sentence-level matching modules adopted in the first step of weakly text-referred target localization, we propose a triad-level matching model concentrating on the triad-proposal similarity, which proves faster and easier to be well trained under the weakly-supervised setting. Despite its simplicity, complex queries can be handled very effectively, as these can be parsed into multiple discriminative triads where the final bounding box results are

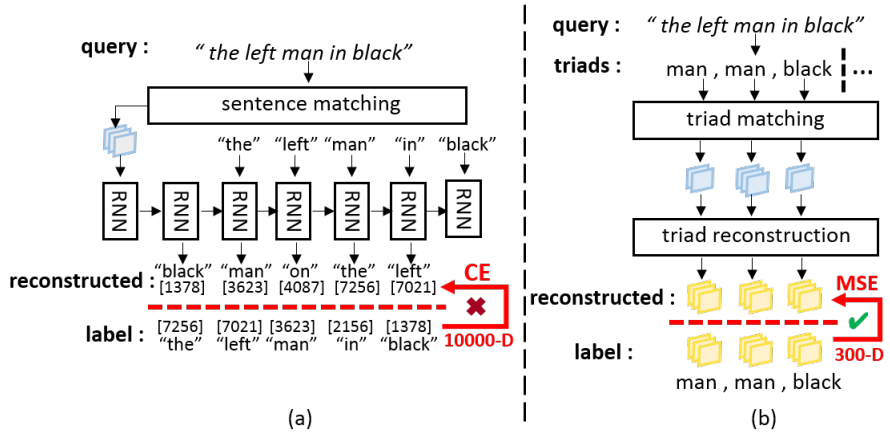

Figure 4.1: Illustration of the difference between (a) the traditional weakly text-referred target localization method, and (b) the proposed triad-level method. The blue and yellow parts indicate the proposal features and the linguistic feature, while the red represents the individual loss function. In (a), although the meaning of the reconstructed sentence is the same as the label, the loss is large, as it is calculated word-by-word, making the network hard to converge. In (b) however, the reconstruction is conducted at the triad level, and a 300-D MSE loss function replaces the 10,000-D CE loss function which dramatically facilitates the weakly-supervised training.

selected by exhaustively considering the matching score of each triad. Accordingly, rather than reconstructing the entire query sentence word-by-word for the second step, we design a triad-level construction module, where the loss is calculated on the linguistic feature of three triad units serving as direct *shortcuts* between the original query and the reconstructed one. This improves the reliability of the loss function and therefore facilitates the weakly-supervised training; see Fig.4.1.(b).

Ultimately, these triad-level matching and reconstruction modules constitute the discriminative weakly text-referred target localization network proposed here that simultaneously provides a light, fast and accurate baseline framework for future weakly text-referred target localization works.

4.1.2 Methodology

Weakly text-referred target localization can be formulated as a region-level retrieval problem. Given an image I, a set of image regions (proposals) $R = \{r_i\}_{i=1}^{N}$ are provided by existing bounding box annotations, or predicted by a region proposal network (RPN) [143], where r is the 4-D vector

denoting the proposal's top-left and bottom-right corners, and N is the number of proposals in image I. The weakly text-referred target localization task is to retrieve the target region r^* according to the query sentence q by maximizing the similarity score $S(r_i, q)$, between all candidate proposals r_i and the query q, that is

$$r^* = _{r \in R} \; S(r, q). \tag{4.1}$$

In this work, the query sentence q is represented by a set of discriminative triads $T = \{t_k\}_{k=1}^{M}$, where M is the number of such triads. Thus, the final similarity score can be calculated as the sum of the individual scores as

$$S(r, q) = \sum_{k=1}^{M} S(r, t_k). \tag{4.2}$$

Consequently, the target is to find a proper scoring mechanism $S(\cdot, \cdot)$ to correctly distinguish the target region from others. Specifically, as the connection annotation for any region-query pair is unknown during training, a reconstructed query $\hat{t}_k$ is predicted from r^* through a reconstruction module, and the training loss is the difference between t_k and $\hat{t}_k$. The whole network architecture is shown in Fig.4.2.

Discriminative Triad

To take full advantage of the discriminative information hiding in queries, we propose to parse and convert a query q into multiple discriminative triads $\{t_k\}_{k=1}^{M}$, each representing a piece of discriminative information to distinguish the target from the distracting objects. A discriminative triad t_k consists of three components, that is, a target unit t_k^t, a reference unit t_k^r and a discriminative unit t_k^d, indicating the target object, the reference object against the target object, and the discriminative relation between them, respectively.

As can be observed from Table 3.1, the discriminative unit t_k^d is defined in a broad sense, which can be extracted from not only subject-relation-object style queries in a straightforward way, such as "the man standing on the table", but also the unary queries containing hidden discriminative connection information, such as "black cat". In addition, some particular tokens are utilized for special referring forms. For instance, the discriminative unit of single-word queries (*e.g.*, "man") is set as "SELF", and "UKN" is used to represent the target unit and reference unit for fragmentary queries (*e.g.*, "left"). Furthermore, complex queries can be separated into several triads. In this way, discriminative triads can represent a diversity of referring forms, no matter whether they are unary, fragmentary or complex.

For its implementation, an off-the-shelf NLP processing toolbox (Stanford CoreNLP[8], Spacy[35])

is employed to analyze the tree structure of each query, as well as the POS tag and the dependency label [22] of each word in the query, to generate discriminative triads. In general, triads are generated within two steps. The first step is to find the target unit shared by all triads. Specifically, the rightmost Normal Noun (NN) of the bottom-left Noun Phrase (NP) in the sentence tree is regarded as the target unit. The second step is to generate the reference unit and discriminative unit for each triad. Phrases in different formations are processed in different parsing patterns. For the unary phrase, such as "the white man" and "the left cat", the first component of the dependency, whose second component is the same as the target unit, except nominal subject (nsubj), prepositional modifiers (prep) and determiner modifiers (det), is viewed as a discriminative unit, with its corresponding reference unit the same as the target unit, indicating a hidden reference object. For the subject-relation-object phrase, such as "the man holding a cat", the second component of the dependency, whose first component is the same as the target unit, and the dependency type is nsubj, is viewed as a discriminative unit, with the second component of the dependency, whose first component is the same as that of the current discriminative unit, is viewed as its corresponding reference unit.

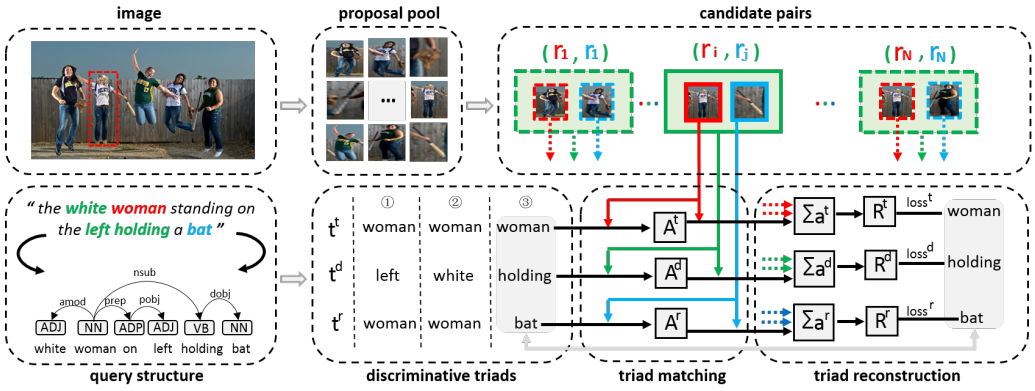

Figure 4.2: The proposed framework mainly consists of the generation of discriminative triads, the triad matching module and the triad reconstruction module (three rightmost modules in the bottom row). Other modules are also illustrated, including the analysis of the sentence structure (leftmost module in the bottom row) and the organization of proposal pairs (modules in the top row). The parts in solid red, green and blue boxes represent the features of a proposal pair used to match the linguistics feature of the target unit, the discriminative unit and the reference unit in a discriminative triad, respectively. Note that the parts in dashed boxes, whose specific routes are omitted, work in the same way as the parts in solid boxes.

Triad-level Matching

The first stage of the triad-level matching is to encode the image I and query q. To encode an image I, given a set of candidate proposals $\{r_i\}_{i=1}^{N}$, I is first passed into a convolutional network (*e.g.*, ResNet [33]), with its last convolutional layer output as the visual feature v_I of image I. The visual feature f_i^v of proposal r_i is cropped within v_I through a pooling mechanism (*e.g.*, ROI pooling [88]). Then, following previous works [133], the spatial feature of each proposal r_i is encoded by a 5-D vector $f_i^s = [\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{w \cdot h}{W \cdot H}]$, where $[\frac{x_{tl}}{W}, \frac{y_{tl}}{H}]$ and $[\frac{x_{br}}{W}, \frac{y_{br}}{H}]$ refer to the top-left and bottom-right corners of r_i, respectively. w and h indicate the width and height of r_i, while W and H denote the width and height of I. In this way, f_i^s contains the absolute position information of r_i, as well as its relative area size to I. To encode a query sentence q with its discriminative triads $\{t_k\}_{k=1}^{M} = \{(t_k^t, t_k^r, t_k^d)\}_{k=1}^{M}$, a pre-trained word vector mechanism (*e.g.*, Word2vec[76], Glove[81]) is employed to extract the linguistic embedding e_k^t, e_k^r, e_k^d with the size of D_l, corresponding to t_k^t, t_k^r, t_k^d, respectively.

The next stage of the triad-matching is to predict the similarity score between q and each candidate proposal r_i in image I through the matching module. As q is parsed and converted into a set of discriminative triads $\{t_k\}_{k=1}^{M}$, the matching module is designed to focus on the single discriminative triad t_k, in order to facilitate the training in the weakly-supervised setting. Furthermore, since an individual discriminative triad t_k comprises target unit t_k^t, reference unit t_k^r and discriminative unit t_k^d, the purpose of the matching module turns to find a suitable proposal pair (r_i, r_j) of I, matching t_k^t, t_k^r and t_k^d based on three similarity scores predicted by the matching module.

Specifically, to predict these scores, three unit-level attention-based matching modules are designed, including the target attention module A^t, the reference attention module A^r and the discriminative attention module A^d. The target attention module A^t outputs a score $a_{k,i}^t = A^t(f_i^v, e_k^t; \Theta_t)$ representing how likely the first proposal r_i in (r_i, r_j) matches the target unit t_k^t, based on

$$a_{k,i}^t = W_2^t \phi_{ReLU}\left(W_1^t \phi_{ReLU}(f_i^v \oplus e_k^t) + b_1^t\right) + b_2^t, \tag{4.3}$$

where W_1^t, W_2^t, b_1^t, b_2^t are parameters of the target attention module weight Θ_t, $\phi_{ReLU}(\cdot)$ indicates the ReLU activation function, and $\oplus$ is a concatenation process.

Similarly, the reference attention module A^r predicts a score $a_{k,j}^r = A^r(f_j^v, e_k^r; \Theta_r)$ representing

how likely the second proposal r_j matches the reference unit t_k^r, as

$$a_{k,j}^r = W_2^r \phi_{ReLU}\left(W_1^r \phi_{ReLU}(f_j^v \oplus e_k^r) + b_1^r\right) + b_2^r. \tag{4.4}$$

Finally, the discriminative attention module $A^d(f_{i,j}^p, e_k^d; \Theta_d)$ predicts the attention score between the entire proposal pair (r_i, r_j) and the discriminative unit t_k^d, according to

$$a_{k,i,j}^d = W_2^d \phi_{ReLU}\left(W_1^d \phi_{ReLU}(f_{i,j}^p \oplus e_k^d) + b_1^d\right) + b_2^d, \tag{4.5}$$

where $f_{i,j}^p$ is a concatenation of $\{f_i^v, f_i^s, f_j^v, f_j^s\}$ containing all information of both r_i and r_j to better extract their interrelated connection information.

Triad-level Reconstruction

Based on the matching scores, a triad-level reconstruction module is designed to rebuild the discriminative triad. The difference between the linguistic feature of the reconstructed triad and the original one is used to optimize the entire network, including both the triad-level matching and the reconstruction modules.

As a single discriminative triad t_k includes three triad units (*i.e.*, t_k^t, t_k^r and t_k^d), the specific reconstruction target turns to rebuild the linguistic feature of each unit, by three corresponding different modules, R^t, R^r and R^d. Two different methods are designed to generate the input of each unit-level reconstruction module, including the soft and hard methods.

In the *soft method*, the input is the weighted sum over the features of all candidate proposals or proposal pairs, aggregated according to corresponding unit-level attention scores:

$$\begin{aligned} \hat{f}_k^t &= \sum_{i=1}^{N} a_{k,i}^t f_i^v \\ \hat{f}_k^r &= \sum_{j=1}^{N} a_{k,j}^r f_j^v \\ \hat{f}_k^d &= \sum_{i=1}^{N}\sum_{j=1}^{N} a_{k,i,j}^d f_{i,j}^p, \end{aligned} \tag{4.6}$$

where $\hat{f}_k^t$, $\hat{f}_k^r$, and $\hat{f}_k^d$ represent the input of the unit-level reconstruction modules, R^t, R^r and R^d, respectively.

In the *hard method*, the input is the feature of the (r_i, r_j) pair with the highest attention score. Since $argmax$ is not differentiable, it is replaced with the differentiable $softmax$ operator through the Gumbel-Softmax [39]

$$\hat{a}^c = softmax(a^c/\tau), \tag{4.7}$$

where $c \in \{t, r, d\}$ is the category of a triad unit, and a^c represents the corresponding attention values calculated by Eqs. (4.3, 4.4, 4.5). τ is a temperature parameter allowing $softmax$ to approache $argmax$ as $\tau \to 0$. $\hat{f}_k^t$, $\hat{f}_k^r$, and $\hat{f}_k^d$ are re-evaluated with the new attention value $\hat{a}^c$ using Eq.(4.6).

After the input is determined by either the soft method or the hard method, contrary to existing works rebuilding the entire query with a heavy RNN-style model, the proposed reconstruction module is designed to reconstruct the linguistic feature of each individual triad unit (*i.e.*, e_k^t, e_k^r and e_k^d) via:

$$\hat{e}_k^c = R^c(\hat{f}_k^c), \tag{4.8}$$

where $c \in \{t, r, d\}$ is the category of a triad unit, $\hat{e}^c$ represents the reconstructed linguistic feature of its corresponding triad unit, and R^c is a lightweight unit-level reconstruction module, which is easier to be trained in the weakly-supervised setting. The triad-based reconstruction module is a simple network, consisting of two fully connected layers with the output size as D_l.

Training and Inference

The entire discriminative network, including the matching module and reconstruction module, is optimized according to the sum of L_2 distances between all original triad units (*i.e.*, e_k^t, e_k^r ,e_k^d) and corresponding predicted ones (*i.e.*, $\hat{e}_k^t$, $\hat{e}_k^r$, $\hat{e}_k^d$). The loss function $\mathbb{L}$ is given by

$$\mathbb{L} = \sum_{c \in \{t,r,d\}} \left\| \hat{e}_k^c - e_k^c \right\|_2^2 . \tag{4.9}$$

During the inference stage, for a certain proposal pair (r_i, r_j), its triad-level attention score $\bar{a}_{k,i,j}$, representing how likely (r_i, r_j) matches the entire discriminative triad t_k, is defined as the weighted sum of three unit-level attentions scores

$$\bar{a}_{k,i,j} = \alpha a_{k,i}^t + \beta a_{k,j}^r + \gamma a_{k,i,j}^d, \tag{4.10}$$

where α, β and γ are the hyper-parameters to control the weight of different attention scores. Then, for a certain proposal r_i, its triad-level attention score $\bar{a}_{k,i}$ is defined as the highest score of

the proposal pairs, where r_i is the first element, as

$$\bar{a}_{k,i} = \max_{r_j \in R} \ \bar{a}_{k,i,j}. \tag{4.11}$$

Ultimately, for a certain proposal r_i, its sentence-level attention score $\bar{a}_i$ is defined as the sum of triad-level attention scores over all discriminative triads through

$$\bar{a}_i = \sum_{k=1}^{M} \bar{a}_{k,i}, \tag{4.12}$$

and the proposal r^* with the highest sentence-level attention score is selected as the predicted result:

$$i^* = _{r \in R} \ (\bar{a}_i). \tag{4.13}$$

4.1.3 Experiments

Implementation Details

The candidate proposals used in this work can be either provided from the bounding box ground-truth in MS-COCO dataset [62], or generated by the detection network, Faster RCNN [88], pre-trained on MS-COCO. Stanford CoreNLP [8] is adopted to parse queries. In the feature encoding stage, the image-level visual feature v_I is extracted through the last convolutional layer of Resnet101 [33]. The individual visual feature of each candidate proposal is cropped within v_I using ROI pooling [88]. Glove word vector [81] is adopted to extract the linguistic feature of the discriminative triad. The discriminative network is trained using one Titan X GPU for around 20 hours adopting the ADAM optimizer [45]. The learning rate is set as 1.3e-5, and the iteration number is set as 150,000 (3 epochs) for all training sets. In the hard reconstruction method, τ is set to initiate the hard reconstruction method. During the inference time, we set (α,β,γ)= (2,1,1) in Eq.(4.10) as the target attention score directly contributes to localizing the target object, while the reference or discriminative attention score only serves as auxiliary information.

Datasets and Metric

The proposed method is evaluated on four common datasets including RefCOCO [133], RefCOCO+ [133], RefCOCOg [73] and RefCLEF [41]. The first three datasets are collected from MS-COCO [62]. The main difference between RefCOCO+ and RefCOCO is that the former one forbids the

Table 4.1: Effectiveness comparison with other weakly text-referred target localization methods. The method with the highest score is in bold. "det" indicates that the candidate proposals generated from Faster RCNN [88]. In other cases, the candidate proposals are provided by the bounding box ground truth.

Method	Setting	RefCOCO			RefCOCO+			RefCOCOg
		val	testA	testB	val	testA	testB	val
VC[90]	-	-	17.34	20.98	-	23.24	24.91	33.79
VC[90]	w/o text-referred target localization	-	13.59	21.65	-	18.79	24.14	25.14
VC[90]	w/o a	-	33.29	30.13	-	34.60	31.58	30.26
VC (det)	-	-	20.91	21.77	-	25.79	25.54	33.66
ARN[67]	-	34.26	36.01	33.07	34.53	36.01	33.75	34.66
ARN[67]	w/o L_{lan}	33.07	36.43	29.09	33.53	36.40	29.23	33.19
ARN[67]	w/o L_{att}	33.60	35.65	31.48	34.40	35.54	32.60	34.50
ARN (det)	-	32.17	35.25	30.28	32.78	34.35	32.13	33.09
KPRN[68]	-	35.04	34.74	36.53	35.10	32.75	36.76	35.44
KPRN[68]	attr	34.93	33.76	36.98	35.31	33.46	37.27	38.37
KPRN[68]	soft	34.43	33.82	35.45	35.96	35.24	36.96	33.56
Proposed	-	**39.21**	**41.14**	**37.72**	**39.18**	**40.01**	**38.08**	**43.24**
Proposed (det)	-	38.35	39.51	37.01	38.91	39.91	37.09	42.54

Figure 4.3: Visualization results on MSCOCO datasets, where the red, green, blue parts represent the target, discriminative, reference unit and their corresponding proposals, respectively.

absolute location words when generating the query sentences, requiring more attention to the appearance discrimination. RefCOCOg contains longer expressions for both appearance and locations. Referring sentences in RefCLEF are annotated casually, so it is normally used as an auxiliary one for validation. The metric used to evaluate the accuracy of the proposed method is similar to the object detection task, that is the Intersection-over-union (IoU) between the predicted bounding box and the ground-truth one is calculated and the one with IoU higher than 0.5 is treated as correct.

Comparison with State-of-the-Art

In this section, to demonstrate the effectiveness and efficiency of the proposed method, and it is compared in terms of accuracy, parameter cardinality and running speed with other SOTA weakly text-referred target localization methods, including KPRN [68], ARN [67] and VC [90].

In terms of the effectiveness, the accuracy comparison is first conducted between the proposed method and other SOTA weakly text-referred target localization methods on three standard datasets, as shown in Table 4.1 and Fig.4.4, including RefCOCO, RefCOCO+ and RefCOCOg datasets. When the candidate proposals are provided from the bounding box ground truth, the proposed method outperforms previous weakly text-referred target localization methods by a large margin on all these three datasets. Specifically, the proposed method boosts the accuracy by 4.17%, 4.08% and 7.8%, against the previous SOTA method, KPRN [68], on the valuation set of RefCOCO, RefCOCO+ and RefCOCOg dataset, respectively. Note that the reason for the higher accuracy gain on RefCOCOg is that queries in that dataset are much longer, where more trivial

words are removed through triad-based parsing. Furthermore, the proposed method achieves a larger accuracy gain over KPRN [68] on the Test A set of RefCOCO and RefCOCO+ datasets, which is 6.4% and 7.26%, respectively. However, the accuracy gain turns smaller on the Test B set of RefCOCO and RefCOCO datasets. We believe the reason is that, compared with the human targets in the Test A set, the general object targets in the Test B set are hard to describe clearly and unambiguously, making it difficult to localize these targets. The rise of accuracy is also seen when the candidate proposals are generated by a detection network Faster RCNN [88].

Apart from the accuracy gain, it can also be observed from Table 4.1 that the proposed method is more stable and robust, compared with other methods. In other words, different from previous methods relying on different settings to achieve the best performance on different datasets, the proposed method performs well on all datasets under a single and fixed setting. For instance, KPRN [68] can achieve its highest accuracy on RefCCOCO dataset under its standard-setting, while for RefCOCO+ dataset, it needs to switch to the *soft* setting to achieve its highest accuracy, and for RefCOCOg, the *attr* setting is required for the best performance. The proposed method, nevertheless, does not need any additional setting, demonstrating its robustness to all datasets. The proposed method is also evaluated on the RefCLEF dataset to further demonstrate its effectiveness. Even with many ambiguous and mistakenly annotated referring expressions in the RefCLEF dataset, the proposed method achieves the accuracy of 34.03%, which is higher than KPRN by 0.17%.

In terms of efficiency, the parameter number of the proposed method (7 million) is approximately 1/3 of ARN [67] and KPRN [68] (24 and 20 million). In addition, the proposed methods run three times faster than ARN [67] and KPRN [68] (0.04 and 0.03 seconds for each picture), demonstrating the efficiency of the proposed system. The major reason is that the heavy RNN-style model is replaced by our proposed model, which is lightweight yet efficient.

Some visualization results of the proposed method are also shown in Fig.4.3, where candidate proposals are provided from bounding box labels. Each diagram, from (a) to (e), illustrates the situation when correspondingly encountering a normal, single-word, fragmentary, reference-hidden or complex query. This demonstrates that the proposed method performs well in a diversity of query formulations.

Ablation Studies

All ablation studies are conducted through cross-evaluation on the training set of RefCOCO, RefCOCO+ or RefCOCOg dataset. Specifically, 3/4 of data of the original training set are used as

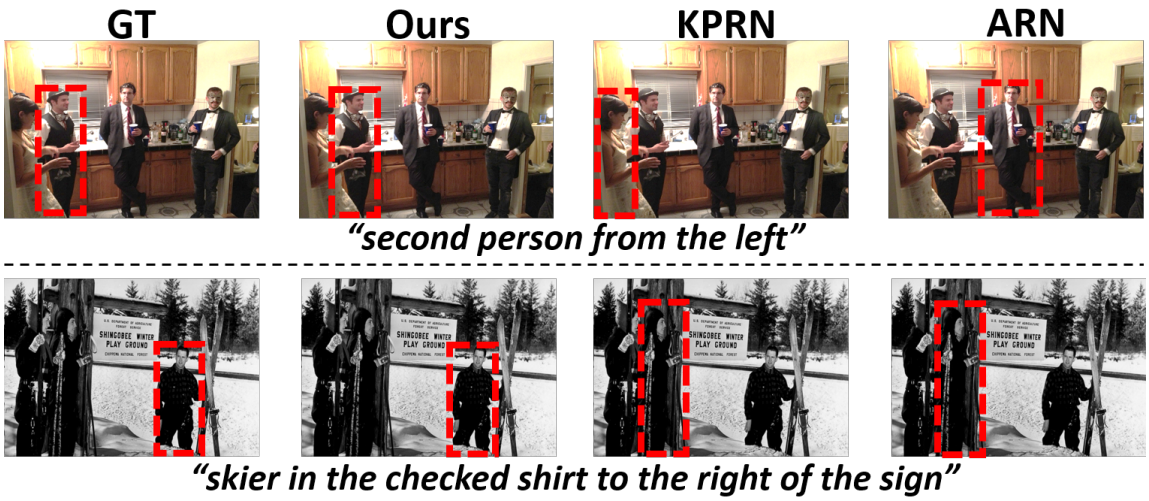

Figure 4.4: The qualitative comparison with other weakly text-referred target localization methods.The red box indicates the bounding box result of the target.

the new training split and the remaining 1/4 of data are used as the new valuation split.

Contribution of Each Unit-level Loss. The first ablation study is to explore the contribution of each unit-level loss component in Eq.(4.9). Specifically, the loss component for the reconstructed target unit, discriminative unit, reference unit is disabled individually during the training stage, generating the setting w/o L^t, w/o L^d and w/o L^r in Table 4.2, respectively. As can be observed from Table 4.2, the accuracy declines considerably if any unit-level loss component is disabled, which demonstrates their joint and individual contribution to the final performance. Furthermore, the largest decline is seen in the w/o L^t setting. We believe the reason is that, the target unit directly contributes to localizing the target object, while the discriminative and reference units only serve as auxiliary information. Based on the aforementioned results, all unit-level loss components are adopted in the proposed method.

Reconstruction Settings. The second ablation study is to explore the best setting for the triad-level reconstruction module. In the first setting, the reconstruction module is disabled, with the feature distance between each (r_i, r_j) pair's feature (*i.e.*, f_i^v, f_j^v, $f_{i,j}^p$), and the corresponding triad unit's linguistic feature (*i.e.*, e_k^t, e_k^r, e_k^d) served as the loss, generating the *w/o Recon* in Table 4.2. The second and third settings are the soft and hard methods described in section 4.1.2, generating *Soft* and *Ours* in Table 4.2. As can be observed in Table 4.2, the none-reconstruction setting *w/o Recon* performs worst, demonstrating the significance of the triad-level reconstruction module.

Table 4.2: Ablation studies through cross-valuation.

Setting	RefCOCO	RefCOCO+	RefCOCOg
w/o L^t	29.5	27.1	27.0
w/o L^d	38.1	36.1	35.5
w/o L^r	37.5	35.7	37.1
w/o Recon	35.6	37.5	35.9
Soft	42.1	39.2	38.2
Single	39.6	38.1	36.1
Ours	**43.2**	**40.1**	**39.8**

In addition, we believe the reason for its bad performance is that f_i^v, f_j^v, $f_{i,j}^p$ and e_k^t, e_k^r, e_k^d are in two different feature domains, and it is arduous to calculate a meaningful distance before feature alignment. For the comparison of the other two settings, as shown in Table 4.2, *Ours* achieves a relatively higher accuracy against *Soft*. The reason is that the Gumbel-Softmax can focus more on the visual feature with the highest attention value and less on other distracting features, facilitating thus the training of the reconstruction modules.

Inference Methods. The third ablation study is to explore the influence of different inference methods. Apart from the standard one described in section 4.1.2 (denoted as *Ours* in Table 4.2), another baseline setting, where a query is represented by a single discriminative triad, randomly selected among all candidate triads, is also evaluated for better analysis, marked as *Single* in Table 4.2. As can be observed from Table 4.2, the bad performance of *Single* demonstrates the significance to take full advantage of all pieces of discriminative information in a query, especially for some complex queries.

4.2 Cycle-free Mechanism

4.2.1 Motivation

As the labels of sentence-region pairs are unavailable during the training period of the weakly text-referred target localization task, existing methods can only adopt a roundabout optimization route based on a cyclic forward-backward pipeline. The purpose of its forward module is to convert a query sentence into an image region. Specifically, a sentence-region matching network is employed to calculate the matching score between the query sentence and each candidate image region, generated through a pre-trained detector (*e.g.,* Faster RCNN [88]), and the image region with the highest matching score is selected out as the result region. Then, the backward module attempts to convert the result region back to the initial query sentence. Based on such a cycle, the difference between the reconstructed sentence and the original sentence serves as the loss to optimize both the forward and backward modules together.

Although this cyclic idea is straightforward, there is still a huge accuracy gap against the fully-supervised methods. We argue that the major limitation comes from the deviation issue which is quite common in cyclic approaches [115]. Such an issue occurs when the selected image region, predicted by the forward module, completely deviates from the target region, but the backward module still reconstructs a similar sentence as the initial query sentence in the backward stage. In this situation, the aforementioned loss function fails to penalize this deviation due to the consistent prediction of the sentence.

Taking Fig.4.5.(b) as an example, the result region generated from the forward module is a region about a cat, which is different from the given query, "dog". In addition, based on the region of the cat, the backward module incorrectly predicts a sentence describing the dog. In this way, both the forward and backward modules make incorrect predictions. However, the loss value in such an error situation is too small to penalize such incorrect predictions, which frequently occur in existing forward-backward weakly text-referred target localization methods.

In this way, the first problem to be addressed is that, such an error situation, where the forward module predicts an incorrect region while the backward module generates the sentence same as the original query, cannot be well penalized in the forward-backward methods. To overcome this limitation, we propose a new cycle-free pipeline for the weakly text-referred target localization task where the aforementioned deviation issue is avoided inherently. Different from the traditional textual-visual matching, as shown in Fig.4.5.(c), the core idea of the proposed method is that the result region can be selected according to the similarity between the initial query sentence and

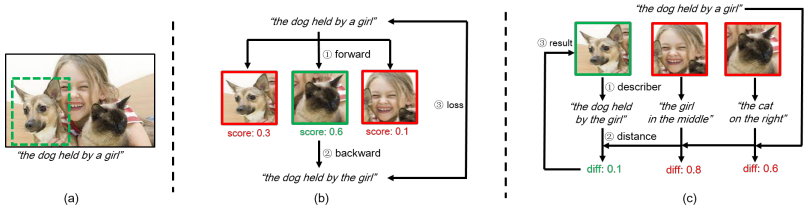

Figure 4.5: The diagram illustrates the difference between the traditional cyclic forward-backward weakly text-referred target localization pipeline and the proposed cycle-free weakly text-referred target localization pipeline: (a) is the given image and query sentence with the target region localized by the green dashed box, (b) is the traditional forward-back pipeline and (c) is the proposed cycle-free pipeline, where "diff" indicates the difference between the query sentence and the textual description. Note that (c) only provides an overview idea of the proposed method, rather than its detailed implementation. Best viewed in colors.

the textual description predicted from each candidate image region. In detail, a region describer is firstly designed to predict a textual description for each candidate region according to the region object's appearance, location and relation information, *etc.* Then, the linguistic feature distance between the original query sentence and each predicted textual description is calculated to measure the similarity. In this way, the image region corresponding to the description with the minimum linguistic feature distance is naturally selected out as the result. Ultimately, such linguistic feature distance can be viewed as a kind of loss function to optimize the region describer network. Thus, two heavy networks (forward and backward) of previous methods are replaced by one light region describer network, and the cyclic optimization route is not existing any longer, with the deviation issue inherently avoided.

To optimize the proposed region describer in the weakly-supervised scenario, among all linguistic feature distances, calculated between all predicted descriptions and the initial query sentence, the minimum feature distance is selected out, serving as the loss to optimize the region describer, with remaining feature distances discarded. In other words, only the sample with the minimum distance is considered as positive and used to optimize the region describer. Such a mechanism encounters an issue that if the initial region describer is not strong enough in the warm-up period, this process can easily drift, with the optimization process collapsed [94].

Thus, the second problem to be addressed is the drifting issue that occurs when the initial network is not strong enough at the beginning of the weakly-supervised learning. To alleviate this problem, a self-paced learning strategy is further employed in the proposed work. The main idea

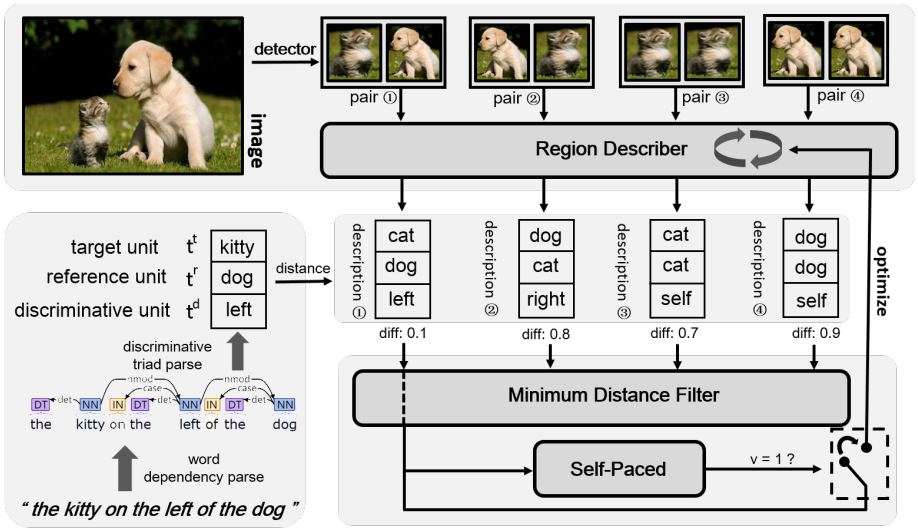

Figure 4.6: The architecture diagram of the proposed method. The upper part is related to the visual information processing, including the proposal generation from the given image using a detection network, the formation of the proposal pairs using the candidate proposals, and the region describer network to generate the textual description of each proposal pair. The bottom left part is about the textual information processing, including the sentence parsing to generate the word dependency and the generation of the discriminative triads. The bottom right part corresponds to the optimization strategy of the proposed method, including the minimum distance filter to select the positive sample with the minimum distance, and the self-paced learning strategy to judge if the filtered sample is suitable to optimize the network. Best viewed in colors.

is that a subset of "easy" samples can be automatically selected by the region describer in each iteration. Training is then performed using only this subset, which is less prone to drift. Later, as the region describer becomes more and more mature after several training iterations, such a subset is progressively expanded with more "hard" samples. Ultimately, all samples can be included for training.

With the proposed region describer and self-paced training mechanism, we aim to provide a new effective baseline framework for future weakly text-referred target localization works.

4.2.2 Methodology

Overview

We formalize weakly text-referred target localization as a region-level retrieval problem, where the result region is selected according to the similarity between the query sentence q and the predicted textual description for each candidate image region.

To localize the target in an image I, a group of image regions (proposals) $R = \{r_i\}_{i=1}^{N}$ are beforehand given by existing bounding box annotations, or generated by a pre-trained detection network (*e.g.,* Faster RCNN [88]) , where r indicates a 4-D vector denoting the proposal's top-left and bottom-right corners, and N is the number of proposals existing in I. Then, the purpose of the weakly text-referred target localization task is to retrieve the target region r^* according to the query sentence q through minimizing the difference between the query q and the predicted textual description of all candidate proposals R:

$$r^* = argmax(q \ominus D(r)), \tag{4.14}$$

where $D(\cdot)$ indicates the proposed region describer which predicts the textual description of each candidate image region in I, and $q \ominus D(r)$ depicts the difference between the query sentence q and the predicted textual description of the region r. Consequently, the target of weakly text-referred target localization further turns to train a robust region describer $D(\cdot)$ which is able to predict correct and discriminative descriptions for all candidate regions, and the aforementioned difference serves as the loss to optimize the region describer $D(\cdot)$. The architecture diagram of the proposed method is shown in Fig.4.6.

Feature Extractor

Two feature extractors are implemented in the proposed method to generate the feature in both visual and linguistic modalities. In terms of the visual feature of the image I, provided with a group of candidate proposals $\{r_i\}_{i=1}^{N}$, I is first passed into a convolutional network (*e.g.,* ResNet [33]), with the output of its last convolutional layer as the visual feature v_I for image I. Then, the visual feature f_i^v of the proposal r_i is cropped within v_I by a pooling approach (*e.g.,* RoI pooling [88]). Then, following previous works [133], the spatial feature of each proposal r_i is encoded by a 5-D vector $f_i^s = [\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{w \cdot h}{W \cdot H}]$, where $[\frac{x_{tl}}{W}, \frac{y_{tl}}{H}]$ and $[\frac{x_{br}}{W}, \frac{y_{br}}{H}]$ refer to the top-left and bottom-right corners of r_i, respectively. w and h indicate the width and height of r_i, while W and H denote the width and height of I. In this way, f_i^s contains the absolute position information of

r_i, as well as its relative area size to I.

To extract the linguistic feature f_q^l, the discriminative triad parsing strategy is employed to highlight the key information of the query sentence q. Specifically, q is firstly parsed into multiple discriminative triads $\{t_k\}_{k=1}^M$, where each triad depicts a piece of discriminative information to distinguish the target object from the distracting or reference objects, and M is the number of triads of q. As shown in Table 3.1, a discriminative triad t_k consists of three individual components, including a target unit t_k^t, a reference unit t_k^r and a discriminative unit t_k^d, denoting the target object, the reference object against the target object and the discriminative relation between them, respectively. Note that, the discriminative unit t_k^d is defined in a broad sense to take full advantage of the potential discriminative information hidden in queries. As for an instance, when it comes to "orange cat" in Table 3.1, this query implies that there are cats in other colors with a high probability. Thus, "cat" is set as its reference unit to represent the distracting cats in other colors.

For the implementation of such a triad-level parsing, an off-the-shelf NLP processing toolbox (*e.g.,* Stanford CoreNLP [8] or Spacy [35]) is adopted to analyze the tree structure of each query sentence, as well as the dependency labels of all words in the query sentence. In general, triads are generated within two steps. The first step is to select the rightmost Normal Noun (NN) of the bottom-left Noun Phrase (NP) in the sentence tree as the target unit shared by all triads. The second step is to generate the reference and discriminative units using a variety of parsing patterns. As for an instance, the subject-relation-object phrase, such as "the man holding a cat", firstly the second component of a particular dependency, whose first component is the same as the target unit, and the dependency type is nsubj, is viewed as a discriminative unit. Then, the second component of another dependency, whose first component is the same as that of the current discriminative unit, is viewed as its corresponding reference unit. Then, a pre-trained word vector generation method (*e.g.,* Word2vec [76] or Glove [81]) is employed to extract the linguistic embedding e_k^t, e_k^r, e_k^d, corresponding to t_k^t, t_k^r, t_k^d, respectively.

With the visual and linguistic features extracted, the next step is to train a region describer network to predict the description of each candidate image region.

Design of Region Describer

In the proposed method, the region describer serves as a bridge between the visual region and the textual description. Then, the linguistic difference between the query sentence q and the textual description of each region can be calculated as the metric to select the result region.

As the query sentence q is parsed into several triads, each of which consists of three units

including a target unit t_k^t, a reference unit t_k^r and a discriminative unit t_k^d, the proposed region describer consists of three corresponding sub-networks as well, including the target describer D^t, reference describer D^r and discriminative describer D^d.

For an image region pair (r_i, r_j), the target describer D^t predicts the target description $\hat{e}_{k,i}^t$ depicting the object information of the first proposal r_i within the proposal pair (r_i, r_j):

$$\hat{e}_{k,i}^t = W_2^t \phi_{ReLU}\left(W_1^t \phi_{ReLU}(f_i^v) + b_1^t\right) + b_2^t, \tag{4.15}$$

where W_1^t, W_2^t, b_1^t, b_2^t are parameters of the target describer, and $\phi_{ReLU}(\cdot)$ indicates the ReLU activation function. Similarly, the reference describer predicts the reference description $\hat{e}_{k,j}^r$ indicating the object information of the second proposal r_j within the proposal pair (r_i, r_j):

$$\hat{e}_{k,j}^r = W_2^r \phi_{ReLU}\left(W_1^r \phi_{ReLU}(f_j^v) + b_1^r\right) + b_2^r. \tag{4.16}$$

Ultimately, the discriminative describer predicts the discriminative description for the entire proposal pair (r_i, r_j):

$$\hat{e}_{k,i,j}^d = W_2^d \phi_{ReLU}\left(W_1^d \phi_{ReLU}(f_{i,j}^p) + b_1^d\right) + b_2^d, \tag{4.17}$$

where $f_{i,j}^p$ is a concatenation of $\{f_i^v, f_i^s, f_j^v, f_j^s, \widetilde{t}_k^d\}$, which contains all information of both r_i and r_j to better extract their interrelated connection information. Note that $\widetilde{t}_k^d$ is a direction vector for the discriminative unit t_k^d of the triad t_k, which is generated from t_k^d through two fully-connected layers. The direction vector $\widetilde{t}_k^d$ is introduced according to an obvious fact that if a query depicts the target according to its color, the describer network needs to predict its color information as well, rather than its location or relation information. Otherwise, the textual difference between the description and query is meaningless. In other words, for a triad t_k of query q and a proposal pair (r_i, r_j), there might be several discriminative descriptions which are all "correct" for the proposal pair (r_i, r_j), $\widetilde{t}_k^d$ can guide the discriminative describer D^d to predict the discriminative description following the same style as t_k^d.

During the inference stage, for each proposal pair (r_i, r_j), the linguistic feature distances $m_{k,i}^t$, $m_{k,j}^r$, $m_{k,i,j}^d$ are calculated firstly as three unit-level scores of each proposal pair, corresponding to the target, reference and discriminative unit respectively:

$$m_{k,i}^t = \left\| \hat{e}_{k,i}^t - e_k^t \right\|_2^2, \tag{4.18}$$

$$m_{k,j}^r = \left\| \hat{e}_{k,j}^r - e_k^r \right\|_2^2, \tag{4.19}$$

$$m^d_{k,i,j} = \left\| \hat{e}^d_{k,i,j} - e^d_k \right\|_2^2, \tag{4.20}$$

where all linguistic feature distances are calculated through L_2 distance metrics. Then, for a certain proposal pair (r_i, r_j), its triad-level score $s_{k,i,j}$, representing whether (r_i, r_j) matches the entire triad t_k, is defined as the weighted sum of three unit-level scores:

$$s_{k,i,j} = \alpha m^t_{k,i} + \beta m^r_{k,j} + \gamma m^d_{k,i,j}, \tag{4.21}$$

where α, β and γ are the hyper-parameters to control the weight of different unit-level scores. Then, for a certain proposal r_i, its triad-level score $s_{k,i}$ is defined as the minimum triad-level score of all proposal pairs, where r_i is its first element:

$$s_{k,i} = \min_{r_j \in R} \ s_{k,i,j}. \tag{4.22}$$

Ultimately, for a certain proposal r_i, its sentence-level score s_i is defined as the sum of triad-level scores over all triads:

$$s_i = \sum_{k=1}^{M} s_{k,i}, \tag{4.23}$$

and the proposal r^* with the minimum sentence-level score is selected as the predicted result:

$$i^* = _{r_i \in R} \ (s_i). \tag{4.24}$$

During the training period, for a certain triad t_k and all proposal pairs, the target describer D^t, reference describer D^r and discriminative describer D^d can be optimized using the minimum distance in $m^t_{k,i}$, $m^r_{k,j}$ and $m^d_{k,i,j}$. To optimize these individual networks in an end-to-end way, we set the overall loss function as the sum of the three aforementioned losses, with all networks optimized simultaneously:

$$\mathbb{L} = \min_{r_i \in R} m^t_{k,i} + \min_{r_j \in R} m^r_{k,j} + \min_{r_i, r_j \in R} m^d_{k,i,j}. \tag{4.25}$$

In this way, the time cost of training can be greatly reduced as the complicated step-by-step training is avoided.

Self-paced Optimization Strategy

As the initial region describer D is not strong enough at the beginning of the training period, the optimization procedure simply relying on Eq. (4.25) tends to drift in the weakly-supervised situation. To alleviate this problem, a self-paced learning strategy is employed in the proposed method to make the optimization procedure robust to the drift issue during the warm-up period.

The core idea of the proposed self-paced learning strategy is to automatically select out easy samples, for which the network is currently more confident, in order to train the region describer D at the beginning. Then gradually more complex samples are used for training when the region describer D is more mature.

Specifically, in a certain optimization iteration, for each training sample (indexed by o), its loss $\mathbb{L}_o(\mathbf{w})$ is calculated using Eq. (4.25) and $\mathbf{w}$ is the parameters of the region describer. A latent weight variable $v_o \in [0, 1]$ is assigned for $\mathbb{L}_o(\mathbf{w})$, indicating whether such a sample is suited to optimize D.

Thus, the purpose of our self-paced learning strategy is to find the proper $\mathbf{w}^*$ and the latent weight $\mathbf{v}^*$ to each sample loss, simultaneously:

$$\mathbf{w}^*, \mathbf{v}^* = argmax\Big(E(\mathbf{w}) + \sum_{o=1}^{C} v_o \mathbb{L}_o(\mathbf{w}) + f(\mathbf{v})\Big), \tag{4.26}$$

where $E(\mathbf{w})$ is weight regularizer for $\mathbf{w}$, C is the number of samples, and $f(\mathbf{v})$ is the self-paced regularizer [136], which determines the specialty of the self-paced learning strategy.

For our proposed method, we implement two kinds of self-paced regularizers, including the discrete and continuous regularizers. When using the discrete regularizer, the value of v can only be discrete (*i.e.,* 1 or 0). The value of v can be continuous when employing the continuous regularizer. The discrete self-paced regularizer is designed as:

$$f(\mathbf{v}) = -\frac{1}{K} \sum_{o=1}^{C} v_o, \tag{4.27}$$

where K is a hyper-parameter ($K > 0$). The closed-form optimal solution of Eq. (4.27) for $\mathbf{v}$ can be written as:

$$v_o = \begin{cases} 1, & \mathbb{L}_o(\mathbf{w}) < \frac{1}{K} \\ 0, & \frac{1}{K} \leqslant \mathbb{L}_o(\mathbf{w}) \end{cases}. \tag{4.28}$$

For the discrete self-paced regularizer, a sample with small loss is viewed as an easy sample, which

should be adopted for optimization, and vice versa. Note that, the value of v_o can be either 1 or 0, making the selection of the training samples too rough to handle complicated situations.

Our continuous regularizer is designed as:

$$f(\mathbf{v}) = -\frac{1}{K}\sum_{o=1}^{C} v_o + \frac{B}{2K}\mathbf{v}^2, \tag{4.29}$$

and its closed-form optimal solution for $\mathbf{v}$ can be written as:

$$v_o = \begin{cases} 1, \ \mathbb{L}_o(\mathbf{w}) < \frac{1-B}{K} \\ \frac{1}{B}\left(1 - K\mathbb{L}_o(\mathbf{w})\right), \ \frac{1-B}{K} \leqslant \mathbb{L}_o(\mathbf{w}) \leqslant \frac{1}{K} \\ 0, \ \frac{1}{K} < \mathbb{L}_o(\mathbf{w}) \end{cases}, \tag{4.30}$$

where $B \in (0, 1)$ is a hyper-parameter. For the continuous self-paced regularizer, v_o is assigned with 1 if the sample loss is too small or assigned with 0 if the sample loss is too large. In other situations, the value of v_o is inversely proportional to the sample loss, making the self-paced learning strategy more flexible to tackle complicated situations. Throughout the chapter, the continuous regularizer is adopted without explicit indication.

4.2.3 Experiments

Implementation Details

The candidate proposals used in this work are generated by the detection network, Faster RCNN [88], pre-trained on MS-COCO [62]. In the feature encoding stage, the image-level visual feature v_I is extracted through the last convolutional layer of Resnet101 [33]. The individual visual feature of each candidate proposal is cropped within v_I using ROI pooling [88]. Stanford CoreNLP [8] is adopted to parse queries. Glove word vector [81] is adopted to extract the linguistic feature of the discriminative triad. The proposed method is trained using one Titan X GPU for around 3 days adopting the ADAM optimizer [45]. The learning rate is set as 6e-6, and the iteration number is set as 10 epochs. The dimension of the direction vector $\widetilde{t}_k^d$ is set as 100. We also set (α,β,γ)= (3,1,1) in Eq. (4.21) since the target description directly contributes to target object localization, while the reference or discriminative description only serves as auxiliary information. For the self-paced learning, K and B are set as 3 and 0.5 initially, and K decays by 0.95 ($K \leftarrow 0.95K$) every 500 iterations.

Table 4.3: Performance comparison with other weakly text-referred target localization methods. "GT" indicates that the candidate proposals are from the bounding box ground truth. In other cases, the candidate proposals are generated from Faster RCNN [88]. "Ave" is the average accuracy of RefCOCO and RefCOCO+ datasets, where the mean accuracy of val, testA and testB sets for each dataset is calculated firstly, through which the final average accuracy for RefCOCO and RefCOCO+ datasets is then obtained. The best score is highlighted in bold, and the second-best score is marked with underline.

Method	Setting	RefCOCO			RefCOCO+			Ave
		val	testA	testB	val	testA	testB	
VC[90]	GT	-	17.34	20.98	-	23.24	24.91	-
VC[90]	w/o a (GT)	-	33.29	30.13	-	34.60	31.58	-
ARN[67]	L_{lan}+L_{adp}+L_{att} (GT)	34.26	36.01	33.07	34.53	36.01	33.75	34.61
ARN[67]	L_{adp}+L_{att} (GT)	33.07	36.43	29.09	33.53	36.40	29.23	32.96
ARN[67]	L_{lan}+L_{adp} (GT)	33.60	35.65	31.48	34.40	35.54	32.60	33.88
ARN[67]	L_{lan}+L_{att} (GT)	38.05	35.27	36.47	34.51	34.40	36.12	35.80
KPRN[68]	hard (GT)	35.04	34.74	36.53	35.10	32.75	36.76	35.15
KPRN[68]	hard + attr (GT)	34.93	33.76	36.98	35.31	33.46	37.27	35.29
KPRN[68]	soft (GT)	34.43	33.82	35.45	35.96	35.24	36.96	35.31
KPRN[68]	soft + attr (GT)	36.34	35.28	37.72	37.16	36.06	39.29	36.98
DTMR[99]	GT	<u>39.21</u>	<u>41.14</u>	<u>37.72</u>	<u>39.18</u>	**40.01**	**38.08**	<u>39.22</u>
Ours	GT	**39.58**	**41.46**	**37.96**	**39.20**	<u>39.63</u>	<u>37.59</u>	**39.24**
VC[90]	-	-	20.91	21.77	-	25.79	25.54	-
VC[90]	w/o text-referred target localization	-	17.14	22.30	-	19.74	24.05	-
VC[90]	w/o a	-	32.68	27.22	-	34.68	28.10	-
ARN[67]	L_{lan}+L_{adp}	31.58	35.50	28.32	31.73	34.23	29.35	31.79
ARN[67]	L_{lan}+L_{adp}+L_{att}	32.17	35.25	30.28	32.78	34.35	32.13	32.83
DTMR [99]	-	<u>38.35</u>	<u>39.51</u>	<u>37.01</u>	**38.91**	**39.91**	<u>37.09</u>	<u>38.46</u>
Ours	-	**39.06**	**39.80**	**37.09**	<u>38.18</u>	<u>39.60</u>	**37.31**	**38.51**

Table 4.4: Ablation studies through cross validation on the RefCOCO dataset. The setting with the best score is highlighted in bold.

Setting	Accuracy
w/o direction	37.6
fixed direction	40.9
w/o self-paced	39.7
discrete self-paced	41.3
w/o triad	35.3
single triad	40.8
Proposed	**42.1**

Datasets and Metrics

Our method is evaluated on two common datasets including RefCOCO [133] and RefCOCO+ [133].

RefCOCO[133]. This dataset is collected from MS-COCO [62]. It comprises 19,994 images, and 142,210 referring expressions with an average query length 3.61. An interactive game (*i.e.,* ReferitGame [41]) is adopted to collect these query expressions. This dataset is divided into four sets including train, validation (val), testA, and testB, with 16,994, 1,500, 750 and 750 images, respectively. The difference between testA and Test B is that, images in testA contain multiple people while images in Test B contain multiple objects of other categories. Note that there is no overlap between these splits.

RefCOCO+[133]. This dataset is also collected from MS-COCO [62]. It contains 19,922 images with 49,856 objects, and 141,564 query sentences with an average length 3.53. These query sentences are collected using the same interactive game as RefCOCO. This dataset is divided into four sets including train, validation testA, and testB, with 16,992, 1,500, 750 and 750 images, respectively. The main difference between RefCOCO+ and RefCOCO is that the former one forbids the absolute location words when generating the query sentences, requiring more attention to the appearance discrimination.

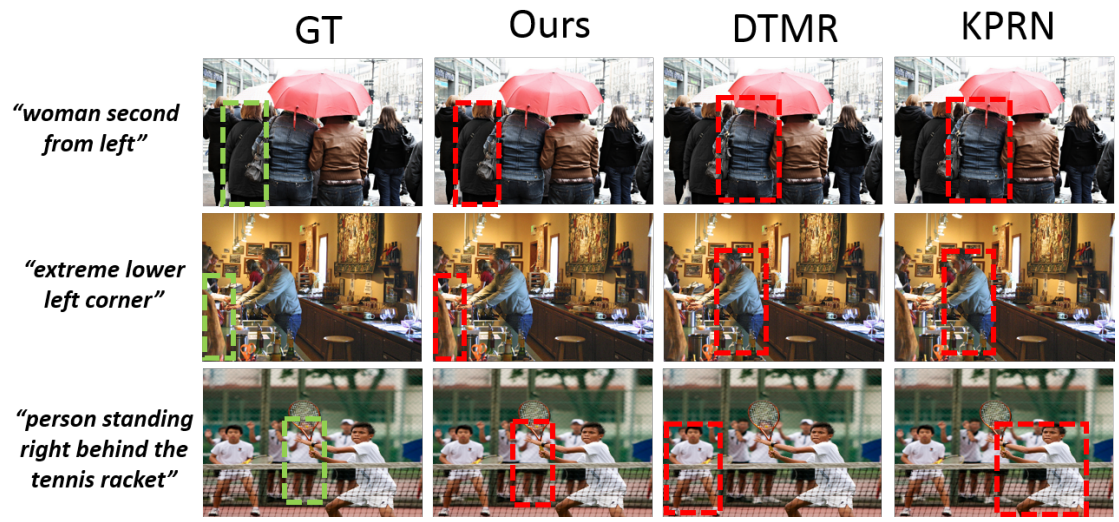

Figure 4.7: The qualitative comparison results between the proposed method and other weakly text-referred target localization methods, where green boxes indicate the labels and red boxes indicate the predictions. Best viewed in colors.

The metrics to evaluate the proposed method are similar to the detection task: the Intersection-over-Union (IoU) between the predicted bounding box and the ground truth one is calculated, and the one with IoU higher than 0.5 is treated as correct.

Comparison with State-of-the-Art

The proposed method is compared with other SOTA weakly text-referred target localization methods, including VC [90], ARN [67], KPRN [68] and DTMR [99], to demonstrate its effectiveness. Qualitative comparison results between the proposed method and other methods are shown in Fig. 4.7, and quantitative comparison results are summarized in Table 4.3. Note that two kinds of evaluation settings are demonstrated, including the label-based and detection-based localization. In the label-based localization, all candidate proposals are provided from the bounding box labels of MS-COCO [62] dataset (marked as "GT" in Table 4.3). While in the detection-based localization, there is no box label provided in advance, and all candidate proposals are generated through the detector network. In the label-based setting, as all candidate proposals are exactly correct, all methods achieve dramatically high accuracy. However, the ground truth bounding boxes

Figure 4.8: Some visualization results of the proposed method: (a) targets correctly localized by the proposed method where the prediction is in the red box, (b) some failed cases of the proposed method where the prediction is in the red box and the ground truth is in the green box. Best viewed in colors.

of candidate objects cannot be provided during the inference period in the real world, which makes the evaluation results under the detection-based setting more meaningful to deal with the practical problems. As can be observed in Table 4.3, the proposed method achieves a higher average accuracy against all existing weakly text-referred target localization methods under both label-based and detection-based settings, demonstrating the robustness of the proposed method.

Thus, the quantitative performance of the proposed method is obtained under the detection-based setting in Table 4.3.

As can be observed, in the detection-based setting, the proposed method achieves new SOTA performances on validation (val), testA and testB sets of the RefCOCO dataset and the testB set of RefCOCO+ dataset against the previous SOTA method [99], which are 39.06%, 39.80%, 37.09% and 37.31%, respectively. In addition, comparable performances are also obtained on other

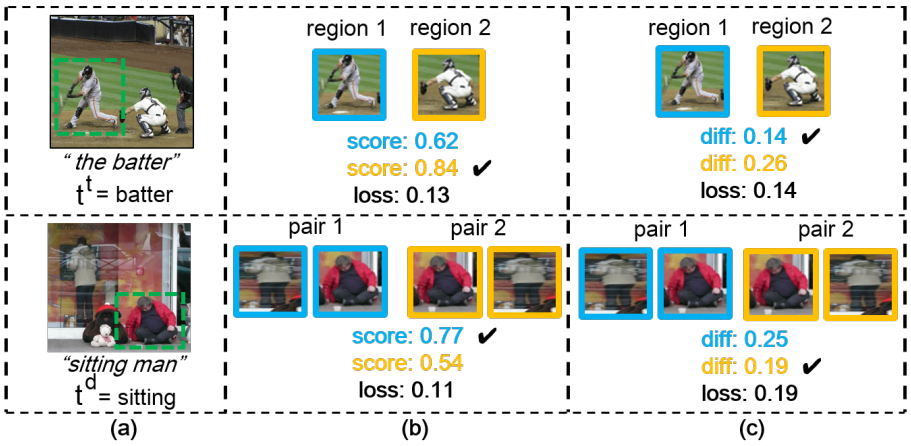

Figure 4.9: Examples of the deviation issue. Column (a) consists of the image, query, target (localized in green box) and a triad unit. The intermediate results of DTMR [99] and the proposed method are illustrated in (b) and (c), respectively. The value in blue corresponds to the region or region pair in blue box, the same as the yellow one. The "score" in (b) is the matching score calculated by the forward module, while "diff" in (c) is the distance between the description and query. Note that unrelated regions are ignored.

sets. Furthermore, the proposed method achieves a higher average accuracy on RefCOCO and RefCOCO+ datasets compared with the previous SOTA method [99], where the mean accuracy of val, testA and testB sets for each dataset is calculated firstly, through which the final average accuracy is then obtained. Such comparison results demonstrate the effectiveness and robustness of the proposed method.

Some visualization results of the proposed method are reported in Fig.4.8.(a), showing that the proposed method performs well in most common situations. However, as can also be observed in Fig.4.8.(b), the proposed method may fail in the following situations: (1) uncommon words exist in the query, (2) the target object is not intact, (3) there are a large number of similar distracting objects in the image. Therefore, more attention should be paid to such hard cases in future works. In addition, as can be observed in Fig.4.9, the deviation issue occurring in the cycle-based method (DTMR [99]) does not exist in the proposed cycle-free method. In both training samples of Fig.4.9.(b), incorrect result region or region pair, with the highest matching score, is selected by the forward module of DTMR [99], while the backward module still predicts a similar sentence with a small loss. Nevertheless, such an issue does not occur in the proposed method as shown in Fig.4.9.(c), where the proposed describer network is optimized with the correct region or region

pair, demonstrating the advantage of the proposed cycle-free method in terms of avoiding the deviation issue.

Ablation Studies

All ablation studies are conducted through cross-validation on the training set of the RefCOCO dataset. Specifically, 3/4 data of the original training set are used as the new training split and the remaining 1/4 data are used as the new validation split.

Influence of Direction Vector. The first ablation study is to explore the influence of the proposed direction vector to guide the region describer to generate the textual description of each candidate region following the description direction of the query sentence (*e.g.,* color, location, relation, *etc*). Firstly, we train the proposed method completely without the direction vector introduced, generating the setting **w/o direction**. Then, we employ the "fixed" direction vector to train the proposed method, generating the setting **fixed direction**. Specifically, for a certain triad parsed from the given query, its discriminative unit is firstly classified into four categories, including "color", "location", "relation" and "other", through a manually-designed dictionary. In such a dictionary, words related to the color, like "red" and "yellow", are grouped as the "color" category; words related to the location, like "left" and "right", are grouped as the "location" category; words related to the relation, like "hold" and "take", are grouped as the "relation" category. The remaining words are grouped as the "other" category. Then, based on the Glove word vector [81], the fixed 300-D word embedding of the category name ("color", "location", "relation", "other") is used as the direction vector for the corresponding triad. As can be observed from Table 4.4, **w/o direction** cannot achieve a comparable result against the settings with the direction vector introduced. The major reason is that under such a setting, the region describer tends to generate a textual description whose description direction deviates from the query sentence. Therefore, the used linguistic feature distance turns meaningless to select the result region. In addition, the proposed setting with the flexible direction vector still performs better than **fixed direction**. We believe the reason is that the manually-designed direction categories cannot cover all directions in reality, leading to lower accuracy.

Influence of Self-paced Learning. The second ablation study is about the influence of the proposed self-paced learning mechanism. We firstly train the proposed method without any self-paced learning mechanism employed, generating the setting **w/o self-paced**. As can be observed from Table 4.4, there is a huge performance gap between the settings with and without the self-paced learning mechanism adopted, which demonstrates the effectiveness of the self-paced

learning mechanism to reduce the drifting issue during the warm-up period. It can also be seen in Table 4.4 that the self-paced learning mechanism based on the continuous regularizer (marked as **Proposed** in Table 4.4) performs better than the one based on the discrete regularizer (marked as **discrete self-paced** in Table 4.4). This is because the self-paced learning mechanism based on the continuous regularizer is able to control the weight of each sample loss in a more flexible and precise way, and the optimal latent weight can be obtained in such a setting.

Sentence Parsing Strategy. The third ablation study is to explore the performance difference of different sentence parsing methods. Apart from the standard one employed in the proposed method, where each query sentence is parsed into multiple discriminative triads, we also train the proposed network without the triad-based parsing, where the linguistic feature of the entire query sentence is directly used to calculate the difference against each region description, generating the setting **w/o triad**. As shown in Table 4.4, an obvious accuracy decline can be seen when adopting **w/o triad**, as it is difficult for the network to successfully learn the linguistic feature space, when the query sentences are full of trivial words and with a variety of patterns. In addition, another baseline setting, where a query is represented by a single discriminative triad randomly selected from all candidate triads, is also evaluated for analysis, marked as **single triad** in Table 4.4. As can be observed from Table 4.4, the relatively lower accuracy of **single triad** demonstrates the significance to take full advantage of all pieces of discriminative information in a query, especially for some complex queries.

Chapter 5

Text-referred Target Tracking

Target tracking and segmentation act as the basis for a multitude of related tasks, including video editing and video surveillance, and typically consist of two steps to process each video sequence: initialization and segmentation. During initialization, target tracking and segmentation are initialized with the first-frame label of the target object, allowing us to focus on the target rather than other distraction objects. Initialization is performed for each new video sequence. During segmentation, the initialized target tracking and segmentation model generates a segmentation mask for the target in the remaining frames of the video.

5.1 Motivation

During initialization, the first-frame target label can be provided in several different forms. Currently, most methods use pixel-level segmentation labels. Based on these pixel-accurate annotations, many methods [110, 137] frequently improve achievable accuracies. However, these methods that are initialized by the segmentation label can be unsuitable for real-world scenarios because generating a pixel-accurate segmentation annotation is time-consuming: it can take 79 seconds to generate a polygon annotation [62], and pixel-accurate segmentation annotation would require even longer. Thus, segmentation labels are infeasible for initialization in real applications, especially when many videos must be processed quickly. To overcome this limitation, [15, 43] investigated alternatives for segmentation labels during initialization and found that language referring expressions [132, 65] is an effective and efficient method, where a sentence, such as "a man in black", is used for interaction between human and computer.

Using language referring expressions, which requires several seconds to generate each anno-

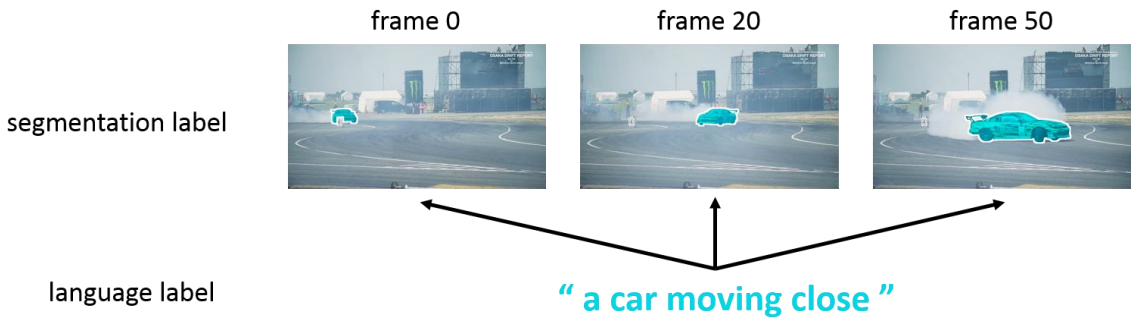

Figure 5.1: A segmentation label is exclusive to only one frame, while a language label is typically compatible with all frames in a video, allowing to select a more suitable frame as the starting point. This video is from the DAVIS 2017 dataset [85], and only one frame is used for language annotation.

tation, provides an adequate replacement for segmentation labels. However, existing methods initialized by language labels [43] achieve lower accuracies than methods initialized by segmentation labels. Considering the possible reasons for these performance gaps, we argue that these language-based methods suffer from three major shortcomings, which reduce their accuracies.

The first drawback of language-based methods lies in the selection of the starting frame for each video. In the era of segmentation mask-based initialization, all methods start from the first frame in a video because the target tracking and segmentation task only provides the first-frame label in a video, preventing the model from starting with any other frames. Conversely, a language label is typically compatible with all frames in a video. As shown in Fig. 5.1, compared with the first frame, starting from the 50th frame facilitates target detection because it provides more detailed object appearance information in the 50th frame. Thus, the final performance can increase, while this positive characteristic of language labels is invariably ignored by existing methods.

Second, the target motion information is never used by existing text-referred target tracking methods. Specifically, most existing methods follow the common pipeline. First, an instance segmentation network (*e.g.*, Mask RCNN [32], YOLACT [21]) is used to generate several candidate proposals and their segmentation predictions. Then, the similarity score between each candidate proposal and the language label is calculated using a matching network. Ultimately, the proposal with the highest score and its segmentation prediction are chosen as the final result. The drawback of this process is that existing methods heavily rely on the matching network trained on the static image dataset [132] and thus fail to extract motion information from a video. Thus, they fail to

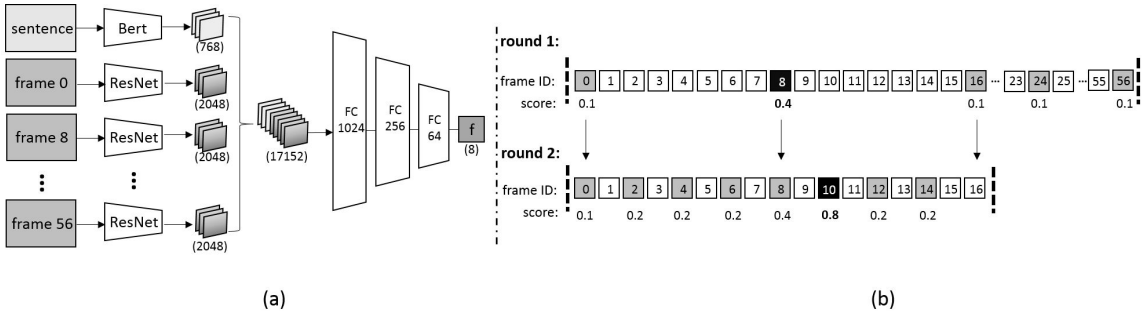

Figure 5.2: Illustration of: (a) the architecture of S-Net ($n = 8$); (b) the way in which n frames are selected in round 1 and round 2, where grids in black or gray indicate the selected frames as the input for S-Net, and grids in black indicate the frame with the highest score predicted by S-Net ($n = 8$).

manage language labels for a moving target (*e.g.*, "a car moving close"), which is common in video sequences.

Third, apart from the similarity score between each candidate proposal and the language label, several other standards are also effective, such as the intersection-over-union (IoU) score or the appearance similarity score between each candidate proposal of the current frame and the prediction result of the previous frame. Some existing methods attempt to use multiple standards to facilitate the matching process in a rough way (*e.g.*, averaging) to combine the scores of different standards [43]. In this study, we use a more flexible method to aggregate these scores to make the best use of the information from different standards, thereby increasing accuracy as much as possible.

There are three motivations for developing the proposed method: 1) the starting frame is fixed as the first one for different videos in existing methods; 2) target motion information is not currently used by existing methods; and 3) there is currently no effective mechanism that considers multiple matching scores.

5.2 Methodology

5.2.1 Overview

The objective of the proposed method is text-referred target tracking, where only the first-frame linguistic annotation, describing the target object, is provided to output the mask for the target

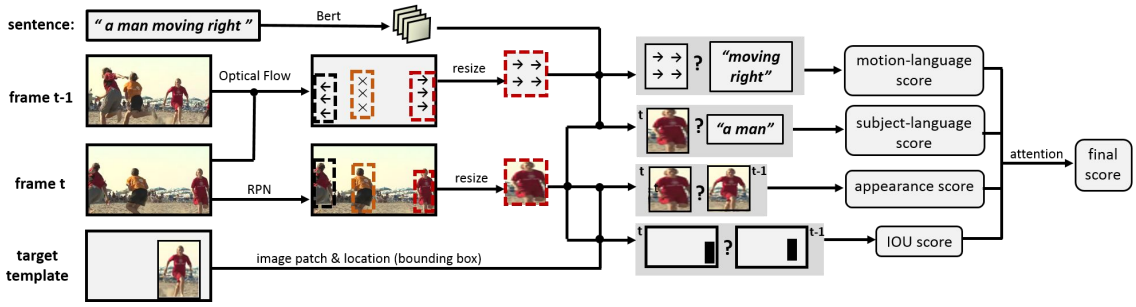

Figure 5.3: Illustration of the segmentation part of the proposed method. First, the motion-language score, subject-language score, appearance score and IOU score for each candidate proposal are calculated through the M-net, visual grounding network, A-net and IOU values. Then, these four scores are aggregated using weights generated by W-net into a final matching score in an attention-based way to determine the most possible proposal corresponding to the target. Foreground segmentation is conducted on this proposal to predict the final target mask (best viewed in color).

object in all frames of a video. The proposed method consists of three steps. For a new video, the first step is to determine the best starting frame within this video for initialization, where the target template is generated, including the target location (bounding box) and image patch within this bounding box. Then, for each frame to be processed, the second step is to calculate a set of matching scores for each candidate proposal, generated by the Region Proposal Network (RPN) [143], according to several matching standards. The third step is to aggregate these scores into a final matching score in an attention-based way to determine the most possible proposal corresponding to the target. Foreground segmentation is conducted on this proposal to predict the final target mask.

5.2.2 Starting Frame Selection

The proposed method takes full advantage of the beneficial characteristic of language annotation. Specifically, a language annotation is typically compatible with all frames in a video; thus, the model can select the starting frame, which is critical to effective initialization. Thus, during initialization, the target template is obtained simply by the similarity score between each proposal generated by RPN and the language annotation. If a wrong target template is obtained during initialization (*i.e.*, the target template refers to another distraction object rather than the target), video object segmentation in the following frames is highly likely to fail due to drift. To a large

extent, this problem can be mitigated by selecting the best starting frame.

Inference

Initially, a language feature extractor is used to generate the feature of the language annotation describing the target, q. Then, n frames with equal intervals are selected among all frames (*e.g.*, frames 0, 3, and 6 are selected from a 7-frame video when $n = 3$). Then, as shown in Fig. 5.2. (a), a visual feature extractor is used to generate the feature of each selected frame r_i, which is concatenated with the language feature as an input for the starting frame selecting network (S-Net) f_s, whose output is a one-hot vector with the size of n, as follows:

$$v_s = Softmax(f_s(q \parallel r)), \tag{5.1}$$

where $Softmax()$ is the softmax function, $\parallel$ is the concatenation process, r is the concatenated feature of all selected frames ($r = r_1 \parallel r_2 \parallel \cdots \parallel r_n$), and v_s is the predicted vector, with each element indicating how much its corresponding frame is suitable to serve as the starting frame. Ultimately, the frame with the highest element value is selected as the starting frame if its value is greater than a threshold T_s. Otherwise, a new sequence around the highest-score frame is removed, as shown in Fig.5.2.(b), where round 2 is conducted via the same procedure as round 1, and the frame with the highest score in round 2 is selected as the starting frame, even if the score is less than T_s.

Training

An existing visual grounding network is used to generate the label for each frame [132]. Each frame in the training set and the target language description is fed into the visual grounding network to predict a region corresponding to the language description. Then, the IOU value between the predicted region and the target's ground-truth region is calculated as the label y_m for this frame. A high IOU value indicates that the target in this frame is easily localized by the language description, and this frame is suitable to serve as the starting frame. In contrast, frames with low IOU values contain some common characteristics, such as motion blur, background clutter, and deformation. Ultimately, S-Net can be optimized through the L2 loss as follows:

$$\mathcal{L}_s = \frac{1}{n} \sum_{i=1}^{n} \|y_i - f_s(x_i)\|^2, \tag{5.2}$$

where x_i is the selected frame and y_i contains the corresponding labels. Note that n frames are selected randomly during training, which is different from the inference process where the interval of every two selected frames is equal, as shown in Fig. 5.2. (b).

Algorithm 1 Pseudo code description of the proposed method.

Language query: gt_l
Sequence length: L
Motion matching network: f_m
Appearance matching network: f_a
Visual grounding network: f_g
Detection network: det
Mask result of frame t: O_t

Extract the linguistic feature q of gt_l.
Generate the optimal starting frame using (5.1), with its index denoted as k.
Generate its mask result O_k simply through the subject-language (SL) score.
Initialize the template $tmp \leftarrow O_k$

for $t = k + 1$ to L **do** Generate candidate proposals in frame t using det.

for each proposal and tmp **do** Calculate the motion-language score using f_m and (5.3).
Calculate appearance score through feature distance using f_a and (5.4).
Calculate the SL score through f_g.
Calculate the IOU score through the IOU value.
Calculate the weights v_m for four scores.
Aggregate four scores.
Select the proposal with the highest aggregated score as the result O_t.
Update the template $tmp \leftarrow O_t$.
Generate mask results from frame $t = k - 1$ to $t = 1$ in the same way.

5.2.3 Matching Score Calculation

Following the selection of the starting frame for target tracking initialization, the next step is to process the remaining frames using both the language annotation and the obtained target template. For each frame, the first step is to feed it into an RPN network to generate candidate proposals for all objects in this frame. Then, according to the target template and the language annotation, each candidate proposal obtains four matching scores: the motion-language score, the subject-language score, the appearance score and the IOU score. These scores are then aggregated

through an attention-based mechanism into a single final score to identify the correct proposal.

Language Similarity Score

The language similarity scores, including the motion-language score and subject-language score, indicate the similarity between the candidate proposal and the language annotation. The motion-language score focuses on the target motion information, while the subject-language score concentrates on the target appearance characteristics.

The subject-language score is calculated by the visual grounding network [132, 65, 129] to take full advantage of the target appearance information. Then, to calculate the motion-language score for a certain candidate proposal p, as shown in Fig. 5.3, the first step is to calculate the optical flow map between the current and previous frames [97]. Then, the bounding box of proposal p is used to localize the region in the optical flow map. The content of this region in the optical flow map, o, contains the motion information of proposal p. Then, the optical flow map, o, and the language feature q of the given sentence are fed into the motion matching network (M-Net) f_m (see architecture in Fig.5.4. (a)) to predict the motion-language score for proposal p, as follows:

$$v_m = Softmax(f_m(o \parallel q)), \tag{5.3}$$

where v_m is the motion-language score. M-Net is optimized using the L2 loss, where the correct proposal is labeled as 1, while the incorrect proposal is labeled as 0.

If the object is static, the referring sentence tends to contain nothing about the motion information, leading to a small weight for the motion-language score, which means that this score does not affect the matching result in this situation. Additionally, if the target object changes its direction frequently or W-Net finds the motion description difficult to manage, W-Net tends to lower the weight of the motion-language score, leaving the matching process to the other three matching branches (subject-language matching, appearance matching, IOU matching); thus, the proposed method works as expected, even if the motion-language matching branch fails by accident.

Appearance and IOU score

For a certain candidate proposal p, its appearance score refers to the similarity between this proposal p and the image patch in the target template. After resizing, an appearance feature extractor, say Resnet [33], is used to generate the feature for the candidate proposal ($feat_a$) and

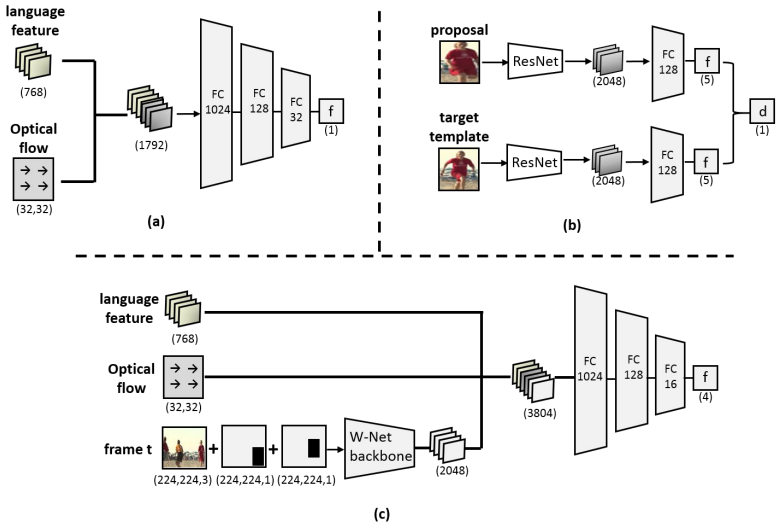

Figure 5.4: Architectural illustration of (a) motion matching network (M-Net), (b) appearance matching network (A-Net), (c) weight generating network (W-Net).

the image patch in the target template ($feat'_a$) as the inputs of the appearance matching network (A-Net) f_a (see architecture in Fig.5.4. (b)) to predict the appearance score. A-Net is a Siamese-style network optimized with contrastive loss [30]:

$$D = \left\| f_a(feat_a) - f_a(feat'_a) \right\|, \tag{5.4}$$

$$\mathcal{L}_a = \frac{1}{2}(1 - y_a)D^2 + \frac{1}{2}y_a \cdot max(0, m - D)^2, \tag{5.5}$$

where D is the distance between $feat_a$ and $feat'_a$, label $y_a = 0$ indicates that $feat_a$ and $feat'_a$ belong to the same object, $y_a = 1$ indicates that $feat_a$ and $feat'_a$ belong to different objects, and m is the margin parameter.

The IOU score is the IOU value between the bounding boxes of the candidate proposal p and the target template. Because the target typically moves slightly between two sequential frames, a high IOU score typically refers to the correct proposal among all candidates.

5.2.4 Matching Score Aggregation

Following the calculation of the four matching scores, the next step is to generate a single final score as the decisive standard to select the correct proposal among all candidates since different matching scores fit different situations. For example, the IOU score is a fast and effective method, especially when the target moves slightly in a video. The appearance score works well when there is a clear appearance distinction between the target and other distraction objects. Language scores can redetect the target if it is lost because the language annotation is typically compatible with all frames in a video. Thus, we propose an attention-based aggregation approach to generate the final score for the matching process. In other words, four attention weights are generated, which indicate the significance of four corresponding matching scores. The proposal with the highest final score is selected and used as the target template. Note that the target template is updated only when the IOU between the selected proposal and the bounding box of the old target template exceeds a threshold T_{IOU}. Otherwise, the target template remains unchanged. Ultimately, this selected proposal is fed into a foreground segmentation network to predict the target segmentation mask.

Inference

For a certain proposal p, during the previous step to calculate four matching scores, the language feature q of the given sentence and the resized optical flow map o localized by proposal p, are already available for the next step. These four scores are merged, as shown in Fig. 5.4. Two one-channel location maps are generated, where the foreground area is labeled as 255, while the background area is labeled as 0. The first location map is the location of proposal p, where the region of proposal p is the foreground, and the other area is the background. The second location map is the target template, where the region in the bounding box of the target template is the foreground, and the other area is the background. These two location maps and the image of the current frame to be processed are then concatenated into a 5-channel input to generate a location-based feature map l. Then, the language feature q, the localized optimal flow map o reshaped into a one-hot vector, and the location-based feature map l are concatenated as the input of the weight generating network (W-Net) f_w to predict four weights v_w for all matching scores, as follows:

$$v_w = Softmax(f_w(q \parallel o \parallel l)). \tag{5.6}$$

Ultimately, with these four attention-based weights, a weighted sum of four matching scores is calculated as the final decisive standard; thus, the proposal with the highest final score is selected as the best-matched proposal.

Training

Before training W-Net, each proposal p is labeled according to whether it corresponds to the target object. Specifically, the IOU value iou_p between each proposal p and the target ground-truth box is calculated. Then, the proposal with the highest IOU value is labeled as 1, while other proposals are labeled as 0. Ultimately, W-Net can be optimized through a modified L2 loss as follows:

$$\mathcal{L}_w = tanh(iou_{max}) \frac{1}{t} \sum_{i=1}^{t} \|y_i - f_w(x_i)\|^2, \tag{5.7}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{5.8}$$

where iou_{max} is the maximum value of all iou_p, t is the number of candidate proposals, x_i is the input proposal for W-Net, and y_i is its label. A scaling factor $tanh(iou_{max})$ is multiplied by the normal L2 loss because RPN can fail to cover the target in the current frame (*i.e.*, none of the candidate proposals corresponds to the target), and this factor is used to minimize the negative effects for such a situation.

Ultimately, the pseudo code of the proposed method is shown in Algorithm 1, including the starting frame selection, candidate mask generation, multiple-standard matching, mask template updating and so on.

5.3 Implementation Details

All networks in the proposed method were trained on an NVIDIA GTX 1080 Ti GPU and a 12-core Intel i7-8700K CPU running at 3.7GHz.

5.3.1 Starting Frame Selection

S-Net was trained with a learning rate of 1e-6. The number of the selected frames n in each round was set as 8. If n was set too large, a much longer time was required to train S-Net due to increasing complexity. If n was too small, S-Net failed to find the best starting frame because the sampled frames cover only a small part of the whole video sequence. Therefore, n was set to 8 as a balance.

The inference process of S-Net was performed within 2 rounds because the video length in the evaluation dataset was typically shorter than 100 frames.

5.3.2 Matching Scores

In terms of the calculation and aggregation of matching scores, the learning rates were set as 1e-6, 1e-5, and 1e-6 for the motion matching network (M-Net), appearance matching network (A-Net) and weight generating network (W-Net), respectively. Edgebox [143] was used to generate the proposal. The off-the-shelf optical flow network and the visual grounding network were trained following [132] and [97]. M-Net and W-Net were trained on the DAVIS 2017 training set [85]. A-Net was trained on the ILSVRC VID 2015 training set [91]. The margin parameter m of the appearance matching network was set to 2, thresholds T_{iou} was set to 0.3 to control the target template update mechanism, and T_s for S-Net was set to 0.6.

5.3.3 Segmentation Network

The segmentation network in the proposed method is a proposal-to-foreground network. Specifically, DeepLab V3 [11] was used as the main part of the proposal-to-foreground network. Its input was an image patch in a proposal generated by the RPN network, and its output was a one-channel probability map with each pixel element indicating the probability that it belongs to the foreground. The atrous convolution layers of the proposal-to-foreground network were pretrained on the PASCAL VOC 2012 training set [26]. Then, the output linear layer was modified into a one-channel layer and trained on the COCO 2012 training set [62] (*i.e.*, each instance object was cropped off in advance). Ultimately, before the evaluation on the DAVIS dataset [83, 85], the method was fine-tuned on the DAVIS training set to adjust to the DAVIS dataset. The learning rate was set as 1e-6.

5.4 Experiments

5.4.1 Experimental Setup

The proposed method was evaluated on four widely-used datasets, including the DAVIS 2017 [85], DAVIS 2016 [83], SegTrack V2 [51] and YouTube-Object datasets [86]. The DAVIS 2016 dataset consisted of 50 high-resolution videos (3,455 frames), where 30 videos were used for training, and 20 videos were used for evaluation. In the DAVIS 2016 dataset, only one single target was annotated

Table 5.1: Quantitative comparison with other related methods on DAVIS 2017 (Da17), DAVIS 2016 (Da16), SegTrack V2 (ST) and Youtube-Object (YO) datasets using the metrics of mean region similarity (J) and the average score of region similarity and boundary similarity ($J\&F$). "Expr" is the task that uses the first-frame language annotation for initialization, and "Mask" is the task that uses the first-frame segmentation annotation for initialization.

Method	Task	Da17$_J$	Da17$_{J\&F}$	Da16$_J$	Da16$_{J\&F}$	ST$_J$	YO$_J$
Ours	Expr	67.2	68.5	83.5	84.8	70.4	79.0
VOSLRE[43]$_{ACCV18}$		37.3	39.3	82.8	-	-	-
Ours-COCO		73.1	75.3	-	-	74.3	79.0
VOSLRE-COCO		45.0	47.5	-	-	-	-
OSVOS[7]$_{CVPR17}$	Mask	-	-	79.8	80.6	65.4	78.3
OnAVOS[111]$_{BMVC18}$		61.0	63.6	85.7	85.0	66.7	77.4
PReMVOS[70]$_{ACCV18}$		73.9	77.8	85.6	86.5	-	-
OSMN[125]$_{CVPR18}$		52.5	54.8	74.0	73.5	-	69.0
FEELVOS[110]$_{CVPR19}$		69.1	71.5	81.5	81.5	-	78.9
DTNet[137]$_{ICCV19}$		64.2	67.4	83.7	83.6	-	-
STMN[79]$_{ICCV19}$		69.2	71.6	84.8	86.6	-	-
AFB[57]$_{NIPS20}$		73.0	74.6	-	-	-	-
DCM[55]$_{NIPS20}$		69.3	72.3	-	-	-	-
GC[53]$_{ECCV20}$		69.3	71.4	-	-	-	-

in each video sequence. The DAVIS 2017 dataset extended the DAVIS 2016 dataset into 90 video sequences (60 videos for training and 30 videos for evaluation). Multiple target objects were also annotated for each video sequence in DAVIS 2017, with an average of 2.03 targets per video and a maximum of 5 targets in a single video. The SegTrack V2 dataset consists of 14 video sequences with large appearance changes and occlusion. The YouTube-Object dataset consisted of 155 video sequences (570,000 frames) divided into 10 groups in different categories, such as planes, birds, and boats. Because the training set and the evaluation set were not differentiated in both the SegTrack V2 and YouTube-Object datasets, all videos in these two datasets were used in the evaluation set. The language annotations of the SegTrack V2 and YouTube-object datasets were collected by the authors and will be released with the source code developed for this study because they are not provided in [43]. Specifically, the target object within the first frame will be annotated. In addition, when there are multiple objects in the video, a detailed language annotation is provided. We describe the target object in terms of its appearance, size, absolute location, relative location and so on to provide sufficient discriminative information to tell the difference between the target object and other distracter objects. The evaluation relies on two common metrics introduced in [85], including region similarity J and contour accuracy F. J is calculated as $J = \left|\frac{m \cap gt}{m \cup gt}\right|$ by IOU between the predicted segmentation m and the ground truth gt. F is defined as $F = \frac{2P_c R_c}{P_c + R_c}$, which indicates the trade-off between counter-based precision P_c and recall R_c using the F-measure.

5.4.2 Comparison with State-of-the-art

Table 5.1 shows that the performance of the proposed method is compared with other text-referred target tracking methods, including VOSLRE [43], OSVOS [7], OnAVOS [111], OSMN [125], FEELVOS [110], DTNet [137], STMN [79], AFB [57], DCM [55] and GC [53]. These methods can be divided into two groups: language annotation and segmentation annotation.

When comparing the methods that use language annotation during initialization, as shown in Table 5.1, the proposed method achieves a region similarity (J) of 67.2% and an average score of region similarity and boundary similarity ($J\&F$) of 68.5%, which are higher than those of VOSLRE [43] by 29.9% for J and 29.2% for $J\&F$, respectively, on the original evaluation set of the DAVIS 2017 dataset. Because the proposed method and VOSLRE use the COCO 2012 dataset [62] to train the segmentation network, the comparison is also conducted on a subset of the DAVIS 2017 evaluation set (43 sequences), where videos with targets that do not belong to 80 COCO categories are excluded (Ours-COCO and VOSLRE-COCO in Table 5.1). In this setting, the proposed method achieves a higher accuracy (73.1% for J and 75.3% for $J\&F$), which is approximately 25% higher

Table 5.2: Ablation study of starting frame on DAVIS 2017.

Setting	J	Δ
Ours	67.2	-
First	63.7	-3.5
Middle	62.6	-4.6
Last	65.1	-2.1

than VOSLRE in the same setting. To compare running times, the proposed method required only 0.4 seconds to process each frame, which is much faster than online updating methods, including OnAVOS [111] (26s) and PReMVOS [70] (37s), and is comparable to methods that do not use online updating, such as FEELVOS[110] (0.5s).

Different from DAVIS 2017, which is an instance-level segmentation dataset, DAVIS 2016 is a foreground segmentation dataset, where multiple objects (*e.g.*, a horse and a man riding the horse) are viewed as one foreground area jointly. Therefore, to evaluate the proposed method on this dataset, a result-merge process is conducted before the prediction of the final result (*i.e.*, candidate proposals with final scores higher than a threshold and their segmentation results generated by the proposal-to-foreground network) are merged to generate the final result for each frame in the DAVIS 2016 dataset. As shown in Table 5.1, compared with VOSLRE, the proposed method increases accuracy by 0.7% for J. When compared with the methods that use first-frame segmentation annotation, the proposed method also achieves competitive performance. Especially on the SegTrack V2 and YouTube-Object datasets, the proposed method even outperforms methods that use the first-frame segmentation annotation, which indicates the robustness of the proposed method on different evaluation datasets.

5.4.3 Ablation Studies

Starting Frame Selection

The first ablation study investigated the significance of the selection of the most suitable starting frame compared with starting from a fixed position for each video sequence. Specifically, the starting frame selection network (S-Net) is disabled. Then, the proposed method is evaluated when

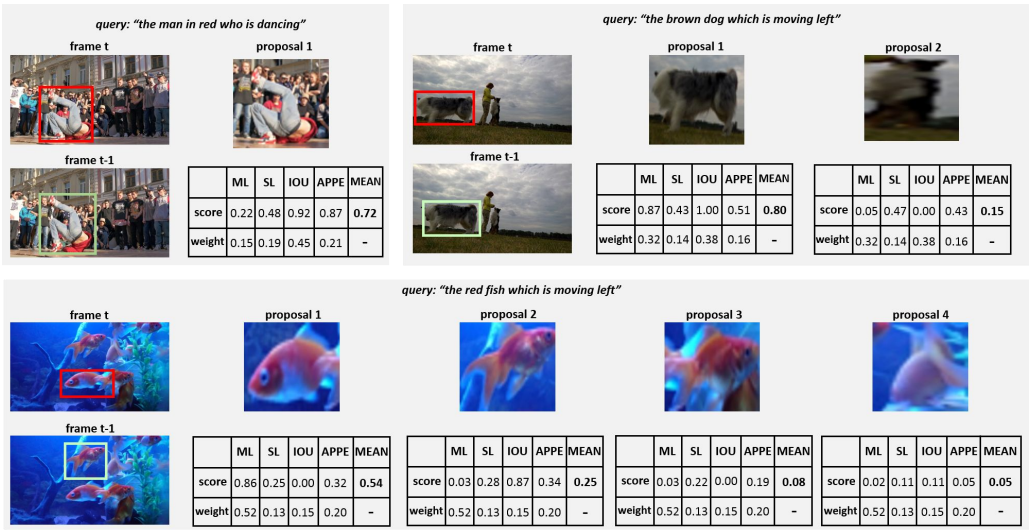

Figure 5.5: In this diagram, the red box in the current frame indicates the correct location of the target, while the green box in the previous frame indicates the predicted target location. ML, SL, IOU, APPE and MEAN are the motion-language, subject-language-matching, IOU, appearance-matching and final average score, respectively. "weight" is the obtained weight value for each standard generated through W-Net. Instead of showing all candidate proposals, only major proposals in a frame are shown in this diagram.

choosing the first frame, the middle frame and the last frame as the starting frame in each video. The results are compared with the proposed method adopting S-Net, and the accuracy difference is listed in Table 5.2, where "First", "Middle" and "Last" describe starting from the first frame, middle frame and last frame for each video, respectively. Δ is the accuracy difference between each setting and the proposed method adopting S-Net. Compared with the proposed method adopting S-Net, accuracy decreases markedly in all fixed-starting-frame scenarios regardless of starting from the first, middle or last frame, which demonstrates the necessity of using S-Net.

Single Standard vs Multiple Standards

The second ablation study investigates the contribution of different matching standards. First, every single standard is individually used during the matching process, with the other three standards abandoned, obtaining "ML", "SL", "APPE" and "IOU" in Table 5.3, indicating that the setting only uses the motion-language, subject-language, appearance and IOU score, respectively. Then, based on the proposed method using all four standards ("IOU+ML+SL+APPE" in Table 5.3),

Table 5.3: Ablation study of matching standards on DAVIS 2017. ML, SL, APPE and IOU indicate the motion-language, subject-language, appearance matching and IOU score, respectively.

Setting	J	Δ
ML	40.3	-26.9
SL	52.1	-15.1
APPE	60.8	-6.4
IOU	38.5	-28.7
IOU+ML	45.7	-21.5
IOU+ML+SL	54.2	-13.0
IOU+ML+SL+APPE (Ours)	67.2	-

Table 5.4: Ablation study on the contribution of motion information on DAVIS 2017 with our new annotations.

Standard	J
Motion-language	50.6
subject-language	44.1

the "APPE", "SL" and "ML" modules are removed in order, obtaining "IOU+ML+SL", "IOU+ML" and "IOU" in Table 5.3. The results show that any single-matching-standard mechanism, as well as the ones with two or three standards, makes an unfavorable comparison against the proposed method adopting all four matching standards, which demonstrates that the four standards are complementary, and all contribute to the final matching process. According to such results, the "APPE" matching score still contributes more to the final result. We believe that the reason for this result is that the visual image can provide more detailed information during matching. Visualization of each standard's contribution is shown in Fig. 5.5, which demonstrates that, for each candidate proposal, the proposed method correctly generates the score for each matching standard, especially the language-based ML and SL scores, while W-Net successfully predicts the importance of each standard. Taking the last case shown in Fig.5.5 as an example, the motion information is more important in this case because only the target is moving left. Thus, a higher ML score and a higher weight value for the corresponding standard are obtained. Finally, using motion information in the query sentence, a good result is achieved in this case.

Subject Information vs Motion Information

The third ablation study investigates the contribution of the motion information used in the proposed method. Table 5.3 shows that when only using the motion-language score during matching, the resulting performance (40.3%) is lower than that of using only the subject-language score (52.1%) because the language annotations provided by [43] contain little description of the target movement, limiting the contribution of the target motion information. Thus, we generate another set of language annotations that consider the description of the target movement. Then, as shown in Table 5.4, when evaluated on the new language annotations, the former achieves a higher region similarity (J) than the latter, which are 50.6% and 44.1%, respectively. Our new language annotations will be released together with the source code.

Segmentation Network

In this ablation study, the segmentation results from existing instance segmentation networks, including Mask RCNN [32] and YOLACT [21], are compared with the proposed proposal-to-foreground network. Both classification labels and segmentation labels in the COCO dataset are utilized when training the instance segmentation networks, while the proposed proposal-to-foreground only uses the segmentation labels for training. The proposed proposal-to-foreground network still achieves a higher region similarity (J) than existing instance segmentation net-

works (66.5% for Mask RCNN and 64.2% for YOLACT on DAVIS 2017), which demonstrates the effectiveness of the proposed proposal-to-foreground network.

Chapter 6

Box-referred Target Tracking

Box-referred target tracking and segmentation methods attempt to rely on the first-frame bounding boxes to provide target object information instead of using the first-frame masks, which dramatically accelerates the annotation process and increases scalability.

6.1 Motivation

This acceleration of box-referred target tracking and segmentation is built on the sacrifice of ground-truth "accuracy". The reason is that some background areas will be incorporated into the bounding box as well, which greatly increases the difficulty. In this way, in order to adapt to the characteristics of box-level ground-truth, most existing methods relying on the first-frame bounding boxes adopt detection-based algorithms. Generally, these detection-based methods include three steps. The first step is to conduct object detection on the whole frame to generate the proposals of all possible objects using the region proposal network [88]. The second step is to do the matching process between the target object and all candidate proposals to find the "correct" proposal. The third step is to do the foreground segmentation on the "correct" proposal to generate the final segmentation result. However, existing methods relying on the first-frame bounding boxes [117, 112] are less performing than the methods relying on first-frame masks both in terms of running speed and accuracy.

First, in terms of the running speed, we observe that most existing detection-based algorithms spend too much time on the matching process, using several time-consuming networks to evaluate the appearance similarity, like the re-identification network [104] and the siamese style network [112]. We observe that, for most video sequences, fast matching according to the intersection

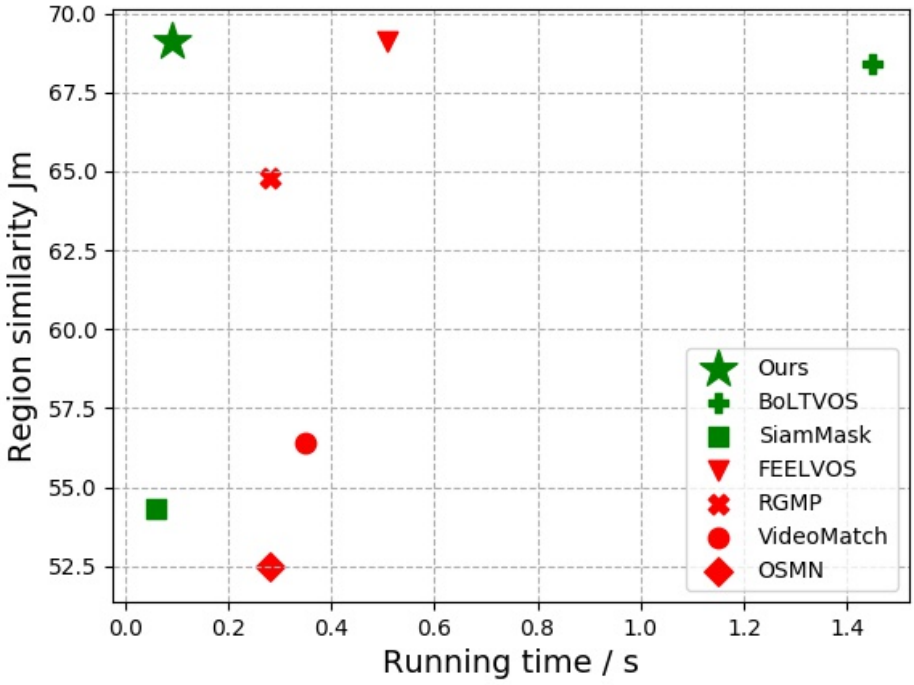

Figure 6.1: Speed-accuracy trade-off for various target tracking and segmentation methods on DAVIS 2017 dataset. Methods in green only rely on the first-frame box-level ground-truth while methods in red rely on the first-frame pixel-level ground-truth.

over union (IOU) between candidate proposals of the current frame and the obtained previous frame's bounding box or segmentation mask can also lead to acceptable performance, because the target object normally moves or changes slowly in two successive frames. However, simple IOU-based matching methods sometimes lose the target, especially when the target disappears out of the sight, and then reappears at a different location. Therefore, neither simple appearance-based matching nor simple IOU-based matching is the best solution for this task.

Second, the major constriction of accuracy is its rough way to update the target template. The target template, containing the target's latest information including appearance, location and so on, plays an essential role in the matching process. Among all candidate proposals of the current frame, the proposal with the highest similarity with the target template will be selected. In this way, whether the correct proposal can be selected is determined by the quality of the target

template as well as its update mechanism. Existing methods, however, simply replace the target template with the newly predicted result after one frame is finished, regardless of the correctness of the obtained result. Therefore, the error will be gradually introduced into the target template, causing a great accuracy decline.

To achieve a better balance between accuracy and speed, a "smart switch" is required to make two significant decisions, including adopting which matching method, IOU-based matching or appearance-based matching, and whether to update the target template or not. To tackle this problem, we formalize it as a conditional decision-making process where only one simple reinforcement learning (RL) agent is employed to make decisions in a flexible way. As can be observed from Figure 6.1, provided with the optimal matching method and updating mechanism, our algorithm can be dramatically accelerated without losing targets even in some difficult frames, leading to a higher accuracy than previous state-of-the-art methods. Specifically, the running speed of our method is approximately 10 times faster than the previous state-of-the-art method.

To sum up, most video object tracking and segmentation algorithms consist of three steps. The first step is to conduct the instance segmentation on the current frame to generate a pool of candidate proposals. The second step is to conduct the matching process to find the correct one as the final result among all candidate proposals according to the target template information. The third step is to update the target template using the prediction of the current frame. In this chapter, as we find the first step does not greatly affect the final result.

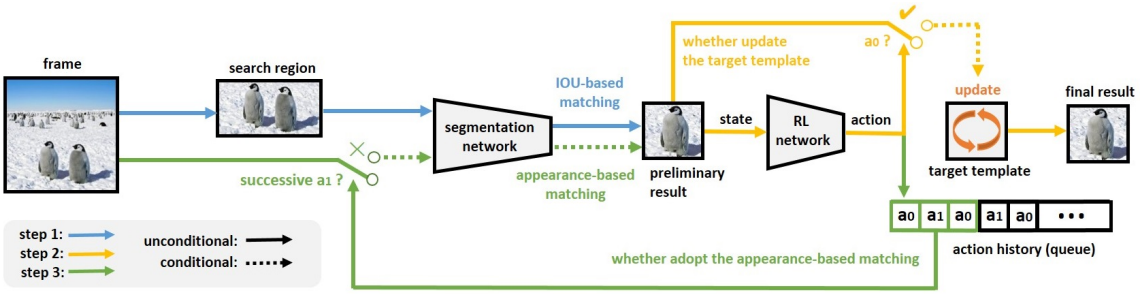

Figure 6.2: Architecture of our method where unconditional paths (full line) indicate they will be conducted in any situation, while conditional paths (dashed line) indicates they will be conducted only in some particular situations.

6.2 Methodology

6.2.1 Overview

Box-referred target localization task only provides the first-frame box-level ground-truth, instead of the first-frame pixel-level ground-truth. The main objective of our work is to utilize RL to boost the performance in terms of both accuracy and running speed, by improving its matching mechanism. To do this, an RL agent is utilized to make two significant decisions simultaneously, including adopting which matching method, IOU-based matching or appearance-based matching, and whether to update the target template or not.

Specifically, as can be observed from Figure 6.2, the processing of the current frame f_t is split into three steps. The first step is to adopt the IOU-based matching to generate a temporary preliminary result. Specifically, the search region b_s of the targets is determined first (see Figure 6.3), which will be fed into a general instance segmentation network (*e.g.* YOLACT [21], Mask R-CNN [32]) to generate numerous candidate predictions. Then, IOU-based matching will be adopted to find the preliminary result among all candidate predictions.

The second step is to determine the update mechanism for the target object information (target template) according to the correctness and quality of the preliminary result, which is judged by the RL agent. If it is good, the target template will be updated by the preliminary result. Otherwise, the preliminary result will be discarded and the target template will keep unchanged. Ultimately, the final result is generated according to the target template.

The third step is to determine whether the appearance-based re-detection is essential for f_t. If the target is lost, in other words, the preliminary result keeps terrible for N successive frames, the target needs to be re-detected again using the appearance-based matching. Otherwise, re-detection for f_t is not needed, and the next frame f_{t+1} will be processed. In terms of the appearance-based re-detection, the whole frame, rather than b_s, will be fed into a general instance segmentation network. Then, a new result will be selected by the appearance-based matching method among all candidate predictions. The second step is conducted again to generate a new final result. Note that the third step is conducted no more than once for each frame.

6.2.2 Agent Action

An RL agent is used to address two complicated challenges during the matching process, which have been ignored by all existing detection-based methods.

In our approach, *target template* is used to represent the target information, which is an

important concept. As shown in Figure 6.4, the target template consists of the target's bounding box T_{box}, segmentation mask T_{mask}, cropped image $T_{box'}$ inside T_{box}, cropped image $T_{mask'}$ inside T_{mask} and the whole frame T_{frame}. Correspondingly, the *predicted result* incorporates the bounding box P_{box}, segmentation mask P_{mask}, cropped image $P_{box'}$ inside P_{box}, cropped image $P_{mask'}$ inside P_{mask} and the current frame P_{frame}.

The first challenge is to decide whether to update the target template using the predicted result. Traditionally, the target template is updated in a rough way, without taking the correctness or quality of the predicted result into consideration. Therefore, when the segmentation network predicts a terrible result for the current frame f_t, which may even refer to another object rather than the target, the target template will still be updated by the incorrect result. This mortal error will make the tracker drift to the wrong target, causing a substantial accuracy drop. Note that this error cannot be avoided by adopting a better matching method because the target was decided before the matching process.

In this way, a "smart switch", which is able to decide whether to update the target template according to the quality of the predicted result, may be the best solution for this challenge. Rather than making the decision heuristically, we adopt an RL agent to make such a decision. The action set A for the RL agent contains 2 candidate actions $a_i \in A$, including a_0 to update the target template using the predicted result of f_t, and a_1 to ignore the predicted result of f_t and keep the target template unchanged.

The second challenge is to decide whether to adopt the fast IOU-based matching method or the accurate appearance-based matching method. In our approach, IOU-based matching views the candidate prediction with the highest IOU score as the correct one, written as:

$$S_{IOU} = \alpha IOU(T_{box}, P_{box}) + \beta IOU(T_{mask}, P_{mask}), \tag{6.1}$$

$$\alpha + \beta = 1, \tag{6.2}$$

where α and β refer to the weight of these two IOUs.

Appearance-based matching views the candidate prediction with the highest appearance similarity as the correct one:

$$S_a = Similarity(T_{box'}, P_{box'}), \tag{6.3}$$

where $Similarity$ function is a siamese style network.

Having two matching mechanisms is inspired by the observation that the fast IOU-based matching performs well for most normal frames, while the appearance-based matching is only

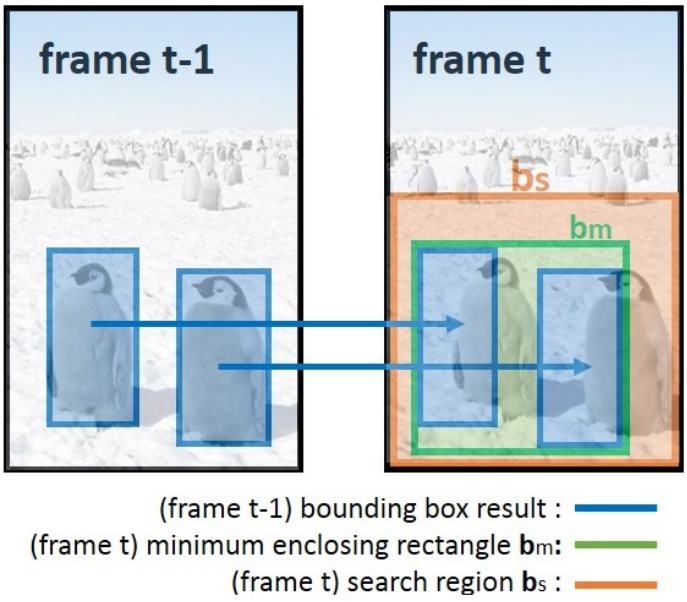

Figure 6.3: Illustration of the minimum enclosing rectangle b_m and search region b_s. b_s is generated by b_m by expansion.

essential for a few difficult situations, especially when the target disappears and reappears again. Therefore, the selection of the matching method is pretty significant to trade off between running speed and accuracy.

Instead of adding one more RL agent, we decide to choose the matching method according to the action history. In fact, the action history intrinsically indicates the prediction quality. If the RL agent predicts a_1 for N successive frames, it is very likely that the target has been lost, and the appearance-based matching is essential to be adopted to detect the target on the whole frame.

6.2.3 State and Reward

The state s_t is the input of the RL agent for frame f_t, including the information assisting the RL agent to predict an optimal action a_t.

In our approach, s_t consists of two parts to provide sufficient information to the RL agent. The first part S_T is the modified image of T_{frame} where $T_{box'}$ remains unchanged while the area outside

T_{box} is blackened, written as:

$$S_T = T_{box'} \cup \Phi(\{i|i \in T_{frame}, i \notin T_{box'}\}), \tag{6.4}$$

where function Φ is to set all pixels black. It provides both the location and appearance information of the target in T_{frame}.

The second part S_P is the modified image of P_{frame} where $P_{mask'}$ remains unchanged while the area outside P_{mask} is blackened:

$$S_P = P_{mask'} \cup \Phi(\{i|i \in P_{frame}, i \notin P_{mask'}\}). \tag{6.5}$$

It provides both the location and appearance information of the predicted objects, as well as its segmentation information.

The ultimate s_t is the concatenation of the feature maps of S_T and S_P:

$$s_t = feature(S_T) \oplus feature(S_P), \tag{6.6}$$

where $\oplus$ indicates the concatenation, as shown in Figure 6.5.

In details, we adopt Resnet-50 [33], pre-trained on the ImageNet classification dataset [23], to extract the feature map of S_T and S_P. We use the first 5 blocks of Resnet-50 [33] which results in a feature map with the size of $\mathbb{R}^{1\times1\times2048}$ for both S_T and S_P, and s_t is with size $\mathbb{R}^{1\times1\times4096}$. Finally, s_t will be fed into the RL agent to predict the action for frame f_t.

The reward function, which reflects the accuracy of the final segmentation result for the video sequence, is defined as $r_t = g(s_t, a_t)$:

$$g(s_t, a) = \begin{cases} 100J_t^3 + 10 & J_t > 0.1 \\ -10 & J_t \leq 0.1 \end{cases}, \tag{6.7}$$

where J_t refers to the region similarity between P_{mask} and the ground-truth mask. Using the cube of J_t expands the difference between the good action's reward and the bad action's reward, which helps to speed up the training of the RL agent.

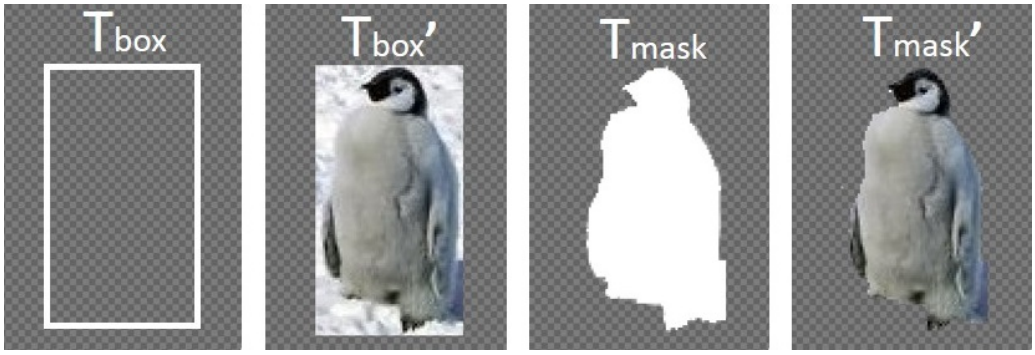

Figure 6.4: Illustration of elements in the target template (T_{box}, T_{mask}, $T_{box'}$, $T_{mask'}$). Elements in the predicted result (P_{box}, P_{mask}, $P_{box'}$, $P_{mask'}$) are in the same form.

6.2.4 Search Region Size

The size of the search region b_s greatly affects the quality of the instance segmentation result. As can be observed from Figure 6.3, b_s is generated using b_m by expansion. The expansion extent of b_m varies according to the video's characteristics. Video sequences with fast-moving targets need to adopt a relatively bigger b_s. Otherwise, targets in these video sequences easily move out of the area of b_s, leading to a terrible segmentation result. Moreover, if some similar targets are close to each other in the video sequence, a relatively smaller b_s is essential so that the instance segmentation network can find the tiny difference between these close and similar targets.

6.2.5 Actor-Critic Training

In our approach, the RL agent is trained under the "actor-critic" framework [46], which is a prevalent RL framework consisting of two sub-networks including an "actor" sub-network to generate the action and a "critic" sub-network to check the quality of this action. Once the RL agent is fully trained, only the "actor" sub-network is used during the inference time.

In our "actor-critic" framework, given a current frame f_t, the first step is to feed the state s_t into the "actor" network and generate an action a_t to decide whether to update the target template using the predicted result. The corresponding reward r_t will also be obtained after conducting this action. r_t is calculated by the region similarity J_t according to (6.7).

Our "critic" network will be trained in a value-based way. Specifically, the parameters are updated as follows

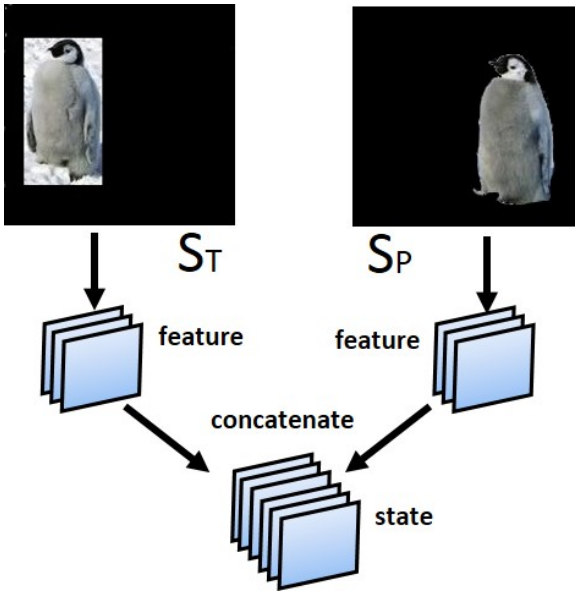

Figure 6.5: The RL state consists of the features of S_T and S_P.

$$w = w' + l_c \delta_t \nabla_{w'} V_{w'}(s_t), \tag{6.8}$$

where

$$\delta_t = r_t + \gamma V_{w'}(s_{t+1}) - V_{w'}(s_t). \tag{6.9}$$

In (6.8) and (6.9), w and w' indicate the weight of the "critic" model after and before update. l_c is the learning rate of the "critic" model. δ_t is the TD error which indicates the difference of the actual score and the predicted score. $V_{w'}(s_t)$ refers to the accumulated reward of state s_t which is predicted by the "critic" model before update. γ refers to the discount factor.

The "actor" network will be updated after the "critic" network in a policy-based way, as follows

$$\theta = \theta' + l_a \nabla (log \pi_{\theta'}(s_t, a_t)) A(s_t, a_t), \tag{6.10}$$

where

$$A(s_t, s_t) = Q(s_t, a_t) - V(s_t) = \delta_t. \tag{6.11}$$

In (6.10) and (6.11), θ and θ' indicate the weight of the "actor" model after and before update. l_a is the learning rate of the "actor" model. $\pi(s, a)$ is the policy function which indicates the

probability of selecting action a in state s. $V(s_t)$ is the score of the state s_t. $Q(s_t, a_t)$ is the score of the state s_t if the action a_t is executed. $A(s, a)$ refers to the advantage function.

In this way, our "actor-critic" framework avoids the disadvantages of both value-based and policy-based methods during the training process. In other words, our RL agent is allowed to be trained and updated at each frame, rather than waiting until the end of the episode, which dramatically speeds up the training process yet maintains training stability.

6.3 Implementation Details

6.3.1 Segmentation Network Training

In our approach, the training of the instance segmentation network follows the strategy of YOLACT [21]. The first step is to pre-train a ResNet-101 network [33] using the ImageNet classification dataset [23]. Then, this network with FPN [60] is used as the feature backbone for the segmentation network. Finally, the segmentation network will be trained on the PASCAL VOC dataset [26] with three losses including the classification loss, box regression loss and the mask loss calculated by the pixel-level binary cross-entropy between the predicted masks and the ground-truth.

6.3.2 RL Agent Training

Our RL agent is trained on the training set of the DAVIS 2017 dataset where all video sequences of the training set are divided into video clips with a fixed number of frames in advance. A video clip, consisting of 10 consecutive frames, is used as an episode for the training of the RL agent. 20 video clips will be randomly selected as a batch. In the beginning, the learning rate l_a for the "actor" model is 1e-4, and the learning rate l_c is 5e-4. l_a and l_c decrease gradually during the training, and they decrease by 1% for every 200 iterations. The discount rate γ for the reward is 0.9. The training of our RL agent takes about 10 days on an NVIDIA GTX 1080 Ti GPU and a 12 Core Intel i7-8700K CPU@3.7GHz.

In terms of other hyper-parameters, when calculating the S_{IOU} in (6.1), we found $\alpha = 1$, $\beta = 0$ for the first frame, and $\alpha = 0.5$, $\beta = 0.5$ for other frames work well, because the pixel-level ground-truth is not available for our task. For the appearance-based matching, we found it is better to re-detect the target using the appearance-based matching when action a_1 is taken for 3 successive frames.

6.4 Experiments

6.4.1 Experiment Setup

We split the experimental evaluation into two sections including the evaluation on tracking and segmentation.

For the tracking experiments, we evaluate our method on four widely-used datasets including DAVIS 2017 dataset [85], DAVIS 2016 dataset [83], Youtube-Object dataset [86], and Segtrack V2 dataset [51]. DAVIS 2016 dataset consists of 50 high quality-videos and 3,455 frames, with 30 videos for training and 20 videos for evaluation. In DAVIS 2016 dataset, only a single target is annotated per video sequence. DAVIS 2017 dataset extends DAVIS 2016 dataset, consisting of 60 video sequences for training and 30 video sequences for evaluation, spanning multiple occurrences of common video object segmentation challenges such as occlusions, motion-blur and appearance changes. In DAVIS 2017 dataset, each video sequence contains 2.03 objects on average, and a maximum of 5 objects to be tracked in a single video sequence. In Youtube-object, there are 155 video sequences and a total of 570,000 frames. All these video sequences are divided into 10 sets according to the category. Youtube-Object dataset does not split the training set and the evaluation set, so we set all video sequences as the evaluation set. Note that Youtube-Object is not an instance-level dataset, in other words, for some videos, several individual targets are annotated into one object as the foreground, which does not completely match our task. Therefore, we split the annotations of these video sequences, so that each target owns its individual instance-level annotation. In SegTrack V2 dataset, there are 14 video sequences with more occlusion and appearance changes compared with Youtube-Object dataset.

Originally, these datasets only provide the pixel-level ground-truth, which does not match our task. Therefore, we generate the bounding boxes according to the pixel-level ground-truth in advance as the first-frame box-level ground-truth.

We valuate our method following the approach proposed in [85]. The adopted evaluation metrics include region similarity J and contour accuracy F. The region similarity is calculated as $J = \left|\frac{m \cap gt}{m \cup gt}\right|$ by the intersection-over-union between the predicted segmentation m and the ground-truth gt. The contour accuracy is defined as $F = \frac{2P_c R_c}{P_c + R_c}$, which indicates the trade-off between counter-based precision P_c and recall R_c using the F-measure.

For the target tracking experiments, we evaluate our method on the LTB35 dataset [47] which is a long-term dataset and was adopted to evaluate the long-term tracking performance in the VOT2018 challenge [47]. LTB35 dataset consists of 35 video sequences with 4,200 frames for each

Method	FT	M	t(s)	D7J	D7$J\&F$	D6J	D6$J\&F$	SJ	YJ
Ours	✘	✘	<u>0.09</u>	**69.1**	<u>70.6</u>	<u>77.5</u>	<u>78.9</u>	**79.2**	**79.3**
BoLTVOS	✘	✘	1.45	<u>68.4</u>	**71.9**	**78.1**	**79.6**	-	-
BoLTVOS *	✘	✘	1.45	60.9	64.9	**78.1**	**79.6**	-	-
SiamMask	✘	✘	**0.06**	54.3	55.8	71.7	69.8	-	-
SiamMask *	✘	✘	0.11	59.5	63.3	75.6	75.9	-	-
MSK(box)	✘	✘	12	-	-	73.7	-	<u>62.4</u>	<u>69.3</u>
FEELVOS	✘	✔	0.51	69.1	71.5	81.5	81.8	-	78.9
RGMP	✘	✔	0.28	64.8	66.7	81.1	81.7	71.7	-
VideoMatch	✘	✔	0.35	56.5	62.4	81.0	80.9	79.9	-
FAVOS	✘	✔	1.2	54.6	58.2	82.4	81.0	-	-
OSMN	✘	✔	0.28	52.5	54.8	74.0	73.5	-	69.0
MSK(mask)	✘	✔	12	-	-	74.8	-	67.4	71.7
PReMVOS	✔	✔	37.21	73.9	77.8	85.6	86.5	-	-
OSVOS-S	✔	✔	9	64.7	68.0	84.9	86.8	-	83.2
OnAVOS	✔	✔	26	61.0	63.6	85.7	85.0	66.7	77.4
CINM	✔	✔	>120	64.5	67.5	83.4	84.2	77.1	78.4

Table 6.1: Quantitative comparison with other methods on the DAVIS 2017 (D7), DAVIS 2016 (D6), SegTrack V2 (S) and Youtube-Object (Y) datasets, measured by the mean region similarity (J), as well as the average score of region similarity and boundary similarity ($J\&F$). **FT** indicates fine-tuning, **M** indicates using the first-frame masks, t(s) indicates the average running time per frame in seconds. The method with the best score is bold, and the method with the second best score is marked in underline.

video sequence on average. In addition, the target will disappear and reappear again for 12.4 times, with an average target absence period of 40.6 frames per video. In this way, this dataset is suited to check the algorithm's ability to re-detect the disappeared target. We evaluate our method following the standard metric for LTB35 dataset. The performance is measured by the F score which is calculated as $F = \frac{2P_r R_e}{(P_r+R_e)}$, where P_r indicates the precision and R_e indicates the recall. Algorithms will be ranked by the maximum F score under different confidence thresholds.

6.4.2 Comparison with State-of-the-art

For the experimental evaluation on the segmentation dataset, we compare our method with other state-of-the-art methods, which are classified into three groups. Methods in the first group only use the first-frame box-level ground-truth, including BoLTVOS [112], SiamMask [117] and MSK [82]. Methods in the second group adopt the first-frame pixel-level ground-truth but do not fine-tune on it, including FEELVOS [110], RGMP [121], VideoMatch [36], FAVOS [17] and OSMN [125]. Methods in the third group fine-tune on the first-frame pixel-level ground-truth, including PReMVOS [70], OSVOS-S [72], OnAVOS [111], and CINM [4]. All quantitative results of the comparison are summarized in Figure 6.1 and Table 6.1. In Table 6.1, the method with the highest score is highlighted and the method with the second best score is marked with underline.

As can be observed from Table 6.1, for the evaluation of the DAVIS 2017 dataset, compared with other methods which only rely on the first-frame box-level ground-truth (BoLTVOS* removes the re-scoring network and SiamMask* adopts the Box2Seg network of BoLTVOS), our method is virtually 15 times faster the previous state-of-the-art method BoLTVOS [112], and our accuracy (mean region similarity J_m) is even higher than BoLTVOS [112] at the same time. For another competitive method SiamMask [117], which runs virtually as fast as our method, our accuracy is much higher than it by around 15%. Even compared with the methods using the first-frame pixel-level ground-truth, our method also outperforms most of these methods. Only PReMVOS [70] achieves higher accuracy than our method, but our method runs 370 times faster than it. For the evaluation of DAVIS 2016 dataset, as it is not an instance-level dataset, our method ranks second among methods in the first group. For the evaluation of SegTrack V2 dataset and Youtube-Object dataset, our methods also achieve a competitive result even compared with methods using the first-frame pixel-level ground-truth.

For the experimental evaluation of VOT, the comparison with other state-of-the-art methods [47] is conducted on the VOT 2018 long-term dataset. As can be observed from Figure 6.6, our method achieves an F score of 0.622, which is quite competitive compared with other methods.

Method	J_m	t(s)
no update	27.3	0.03
simple update (IOU)	63.1	0.03
simple update (appearance)	67.2	1.10
RL update (IOU)	68.1	0.06
Ours	**69.1**	0.09
Ours (supervised)	60.2	0.09
BoLTVOS	68.4	1.45
SiamMask	54.3	0.06

Table 6.2: Ablation studies on the DAVIS 2017 dataset, measured by the mean region similarity (J_m).

Overall our method achieve very good speed-accuracy trade-off. This result also shows our method is able to handle both target tracking and segmentation, even in the situations where the target disappears and reappears again at a different place.

6.4.3 Ablation studies

Contribution of each component: We conduct the ablation studies on the DAVIS 2017 dataset [85], where parts of our methods are disabled to investigate the contribution of each component. The results of the ablation studies are reported in Table 6.2.

First, we totally remove the target template update mechanism, obtaining the method **no update**, which causes numerous drift, and leads to a terrible accuracy (27.3%). Then, we evaluate the simple update mechanisms, where the target template will always be updated, and only IOU-based matching or only appearance-based matching is adopted to select the predicted result, obtaining method **simple update (IOU)** and method **simple update (appearance)** respectively. As can be observed from Table 6.2, only adopting IOU-based matching already achieves an acceptable accuracy (63.1%). Although the accuracy of method **simple update (appearance)** is 4.1% greater than method **simple update (IOU)**, the sacrifice in speed is unacceptable (from 0.03s to 1.1s), which proves the inefficiency of the simple target template update mechanism, where

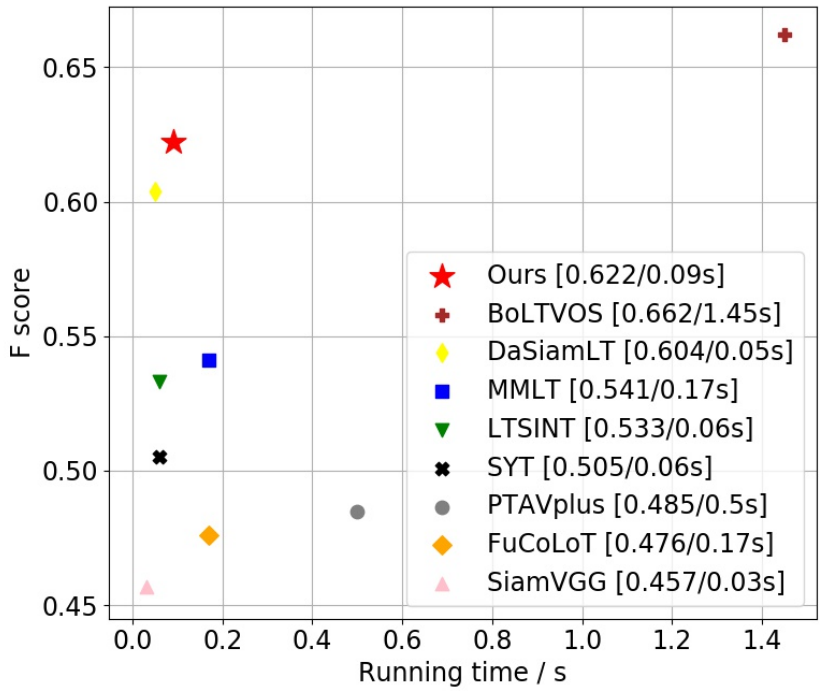

Figure 6.6: Speed-accuracy trade-off on the VOT 2018 Long-term dataset [47].

only appearance-based matching is adopted. In addition, the accuracy of method **simple update (appearance)** is still lower than that of our overall method, which demonstrates that the target template update issue cannot be totally solved simply by adopting an accurate matching method. Finally, we evaluate our method without the usage of appearance-based matching, whose accuracy is 1.0% lower than that of our overall method. This gap is not big, because the situation where the target disappears and reappears from another place is pretty rare in the DAVIS 2017 dataset.

RL or supervised learning? Apart from training the network with RL, we also attempt to train the network in a supervised way, *i.e.* evaluating the fixed label for each frame in advance before training, but finally, as can be observed from Table 6.2, we find the model trained under the reinforcement learning way performs much better than the model trained under the supervised way, which achieve the accuracy of 69.1% (RL) and 60.2% (supervised). We believe the major reason is that, an RL model considers not only the current profit but also the potential profit in the future,

due to the adopted accumulative future reward for training. In other words, the model trained in a supervised way tends to be myopic, while the model trained in the RL way pays more attention to the global and overall performance, making it more suited to video-related tasks.

Choice of instance segmentation: Apart from adopting the fast one-stage segmentation network, like YOLACT [21], we also attempt to adopt the slow two-stage segmentation network, like Mask R-CNN [32]. Then, the running speed of our method drops slightly from around 90ms to around 150ms, and it is still around 10 times faster than the previous state-of-the-art method, BoLTVOS [112], whose running speed is around 1.45s. In addition, as Mask R-CNN achieves higher accuracy than YOLACT [21], our final accuracy is even slightly higher than the proposed one. Note that the choice of the instance segmentation method does not greatly affect the final result, both for running speed and accuracy.

Chapter 7

Mask-referred Target Tracking

For mask-referred target tracking and segmentation, the ground-truth annotation for each pixel of the first frame is provided. The segmentation model identifies pixels of foreground and background for the following frames.

7.1 Motivation

Most recent approaches addressing semi-supervised video object segmentation are based on convolutional neural networks. In particular, the one-shot video object segmentation (OSVOS), introduced by Cacells *et al.* [7] has achieved great success. The core idea is to fine-tune an ImageNet [23] pre-trained ConvNet on the video object segmentation data-set, such as DAVIS [83, 85], to allow the segmentation model to find general objects in a frame at first. Then, the first frame of the inference video sequence will be used to fine-tune the segmentation model such that it can rapidly focus on the specified target object instance in the first frame. Despite the promising result obtained using OSVOS, such a method is challenged by the cases where the appearance of the target object changes dramatically in the video sequence and there may exist multiple confusing instances of similar appearance. OSVOS only learns the target object appearance from the first frame of a video sequence. It cannot adapt to the target object appearance variation when deformation occurs or the camera rotates.

Inspired by the fact that the online adaption has achieved great progress on video object tracking at bounding box level
[120, 69, 31, 18, 6, 20, 122], some works start to deploy the online adaption to mask-referred target tracking and segmentation. For instance, OnAVOS [111] proposes an online adaptive video

object segmentation method that enables to update the segmentation model during inference time. However, model drift may interrupt the model online adaptation and lead to a performance drop.

We observe that, for online mask-referred target tracking and segmentation model update, how to identify the ROI for model adaptation is critical. In particular, as can be observed in Figure 7.1, if we select the adaptation area in a rough way, the segmentation model after adaptation performs very poorly when there are multiple similar objects in the frame, especially, the distraction object is very close to the target. As such selecting the optimal adaptation area in a more sophisticated and flexible way is very significant for video object segmentation.

To tackle this problem, we formalize mask-referred target tracking and segmentation as a conditional decision-making process where two reinforcement learning (RL) agents are employed to infer and adjust the ROI for adaption for both foreground and background in a flexible way. At each frame, the RL agent outputs actions to tune the ROI. Provided with such regions, the mask-referred target tracking and segmentation model can be updated to be more specific and discriminating for the instance in the current frame.

To select the optimal adaptation area for video object segmentation, a set of features of different adaptation areas of the current frames will be fed into the RL model. Then, the RL model will select the best action, and choose the most suitable adaptation area for the current frame. As a result, the segmentation model can obtain an accurate segmentation result. Though the RL method is promising for region identification, it is notoriously slow to optimize the agent. In this work, to speed up the RL agent training, we propose a multi-branch tree-based policy search method where possible action state pairs are organized in a tree structure.

7.2 Methodology

7.2.1 Overview

In general, the main objective of our work is to utilize RL to improve the performance of video object segmentation with online adaptation. Different from the existing approaches to select the adaptation ROI in a rough way, our work utilizes RL to choose the optimal adaptation ROI for each individual frame to avoid model drift. In other words, different frames will own their particular standard to choose adaptation ROI. The overview diagram of our proposed method is described in Figure 7.2.

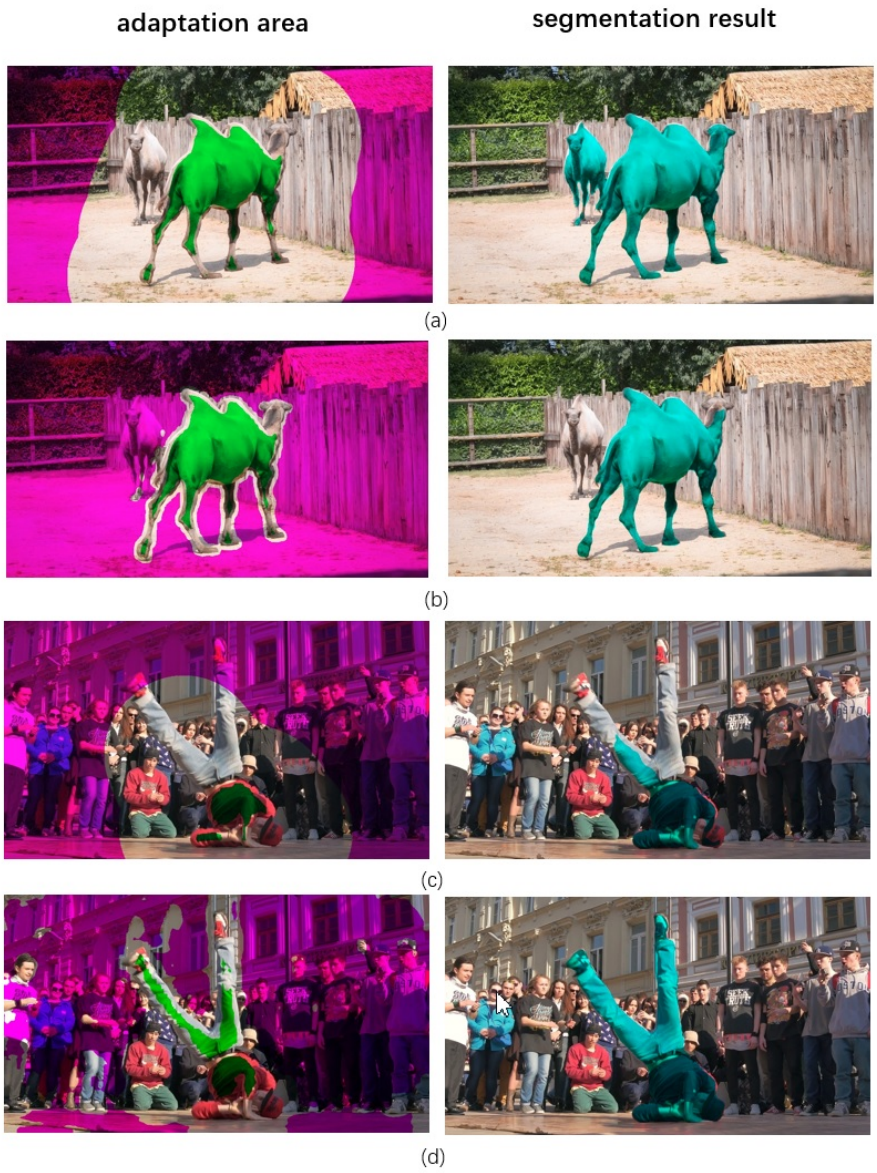

Figure 7.1: Different adaptation areas lead to different frame segmentation results. The segmentation models are the same before online adaptation. In the left column, the area in green is the adaptation area for foreground and the area in purple is the adaptation area for background.

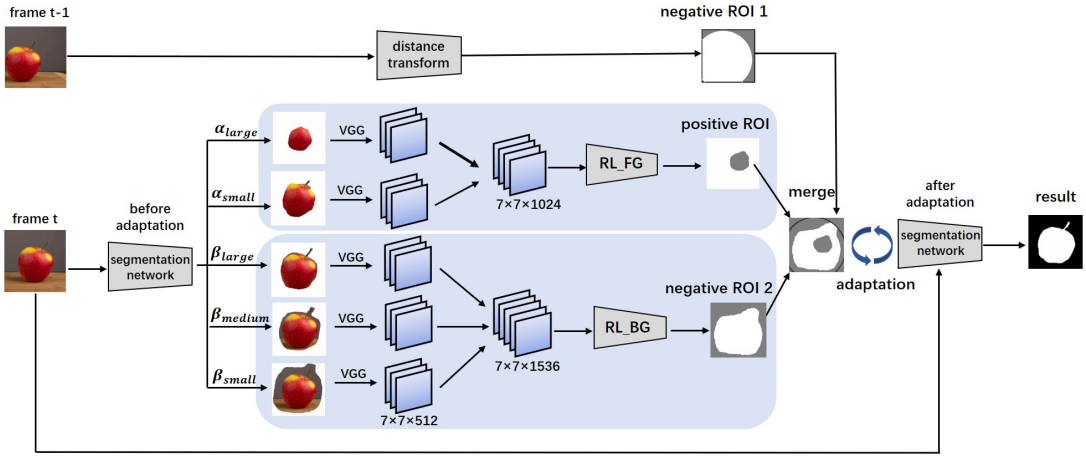

Figure 7.2: The network architecture of our work consists of two RL models. One is to choose the adaptation area for the foreground, and another is to choose the adaptation area for the background. We use pre-trained VGG19 model to extract the feature of each ROI area of the current frame, and then feed the combined feature into the RL model as the state. Finally, the RL model will choose the best adaptation area and update the segmentation model using the chosen adaptation area.

Online Adaptation Mechanism

Our work is built on the top of the model with online adaptation. Before finally segmenting the current frame F_t of a video sequence, a part of pixels in F_t will be used to update the current segmentation model to adapt to the change of the foreground and background. These pixels consist of positive ROI regarded as foreground, denoted as S_p, and the negative ROI regarded as background, denoted as S_n. In OnAVOS [111], S_p and S_n are chosen according to two fixed thresholds including T_p and T_n. In order to determine S_p, F_t will be fed into the segmentation model and a temporary (before online adaptation) probability map M_f, whose size is the same as the F_t, will be generated. In M_f, for a pixel i, $M_f(i)$ refers to the probability that pixel i belongs to the foreground of F_t. Then, S_p will be chosen according to T_p, as follows:

$$S_p = \{i | i \in F_t, M_f(i) > T_p\}. \tag{7.1}$$

Negative ROI S_n is decided by the distance to the positive ROI S_p. S_n indicates the pixels far away from S_p, as follows:

$$S_n = \{i | i \in F_t, distance(i) > T_n\}, \tag{7.2}$$

where $distance(i)$ refers to the distance of pixel i to S_p.

RL Based Online Adaptation

Existing methods with online adaptation, *i.e.*, OnAVOS [111], adopt a fixed standard to select the adaptation area, including T_p and T_n, where different characteristics of each frame are not considered. To address this problem, we use flexible thresholds, t_p and t_n, to choose the adaptation ROI. We build two RL models to choose the most suitable t_p and t_n for each frame.

Our RL framework includes state $s \in S$, threshold selection action $a^p \in A^p$ to determine the value of t_p and threshold selection action $a^n \in A^n$ to determine the value of t_n, state transition function $s' = T(s, a^p, a^n)$ and the reward function $g(s, a^p, a^n)$. Given a frame F_t of one sequence, first of all, F_t will be fed into the segmentation network to obtain a temporary probability map M_f. A set of ROIs can be obtained according to the candidate thresholds. In our work, we need 5 ROIs where the probability value is greater than α_{large}, α_{small}, β_{large}, β_{medium} and β_{small}, respectively. Note that β_{micro} is ignored to diminish the complexity of the state. The first two ROIs are used for the RL model to choose t_p while the last three ROIs are used for RL model to choose t_n. In other words, the possible values of t_p can be α_{large} or α_{small}, and the possible values of t_n can be β_{large}, β_{medium}, β_{small} or β_{micro}. After t_p and t_n are determined, pixels with probability values greater than t_p will be viewed as the adaptation ROI for foreground, as follows:

$$S_p = \{i | i \in F_t, M_f(i) > t_p\}, \tag{7.3}$$

while these pixels with probability value less than t_n, combined with the negative pixels chosen by distance using (7.2), will be regarded as the adaptation ROI for background, as follows:

$$\begin{aligned} S_n = \{i | i \in F_t, distance(i) > T_n\} \\ \cup \{i | i \in F_t, M_f(i) < t_n\}. \end{aligned} \tag{7.4}$$

S_p and S_n are used to update the current segmentation model while other pixels are ignored. After the segmentation model has been updated, F_t will be fed into the new segmentation network and the final segmentation result is obtained.

The pseudo-code of our algorithm is described in Algorithm 2.

Algorithm 2 RL Based Video Object Segmentation

Ground-truth of the first frame $gt(1)$
Sequence length L
Distance threshold T_n
Segmentation network Seg_Net
Pretrained VGG netwrok VGG
RL model to choose positive threshold RL_p
RL model to choose negative threshold RL_n
Segmentaion result of Frame t O_t

Fine-tune Seg_Net on F_1
$last_mask \leftarrow gt(1)$

for $t = 2$ to L **do** Obtain 2 RL states using (7.5) and (7.6), respectively.
Feed the states into the RL models and achieve the optimal threshold t_p and t_n.
Obtain positive ROI and negative ROI using (7.3)(7.4), respectively.
Update Seg_Net on F_t using S_p and S_n.
$O_t \leftarrow forward(Seg_Net, F_t) > 0.5$
$last_mask \leftarrow O_t$

7.2.2 Agent Action

The framework of our work consists of two RL models, including one to choose t_p and another to choose t_n, as shown in Figure 7.3. The action set A^p used for the first RL model to choose S_p contains 2 candidate thresholds: α_{large} and α_{small}. As the difference of the ROI areas with different t_p is not very big, we only set two candidate thresholds for the action set A^p. The action set A^n used for the second RL model to choose S_n contains 4 candidate thresholds: β_{large}, β_{medium}, β_{small} and β_{micro}. As the size of S_n is much larger than that of S_p, we set more candidate thresholds for t_n.

In terms of the design of the candidate values of t_p, in OnAVOS [111], T_p is 0.97 and it is fixed for all frames. This threshold is very safe and conservative because it should work in any situation, especially for some frames with very bad segmentation results. In fact, for some frames with good segmentation results, the value of t_p ought to be much lower, so more correct pixels of the object can be used to update the model and improve the final segmentation result. As the difference in the size of the adaptation ROIs with different t_p is not very huge, we only set two candidate thresholds for the action set A^p, which can lower the difficulty of training the RL model.

In terms of the design of the candidate values of t_n, in OnAVOS [111], S_n are chosen according

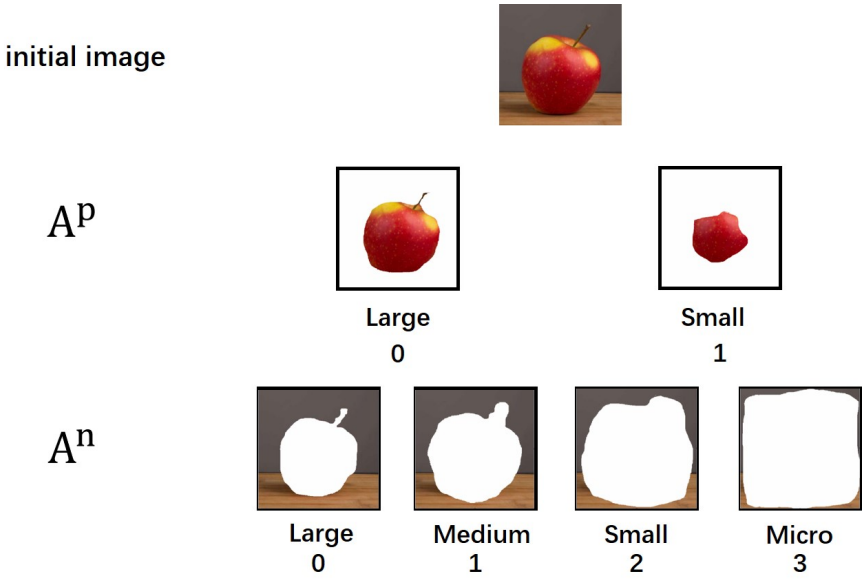

Figure 7.3: Foreground adaptation ROI selecting action set A^p and background adaptation ROI selecting action set A^n.

to the distance to the target object rather than the value of the M_f. If a pixel is far away from the target, it will be viewed as S_n. Similarly, the selection of S_n in OnAVOS [111] is very conservative as it works for almost all frames. In our work, we still keep the adaptation ROI which is chosen by the distance, and then try to add more areas to include more correct adaptation ROI for background. The additional area is chosen by the value of M_f. Finally, we combine these two parts of areas together as the final area of S_n using (7.4). As the size of the area for S_n is much larger than the area of S_p, we set more candidate thresholds for t_n.

In this way, after training, these two RL models can choose the best action and achieve the optimal thresholds t_p and t_n.

7.2.3 State and Reward

The state s is the input of the RL model. As we have two RL models, we need two sets of states for two models. In general, the state s is a feature map combined with the feature maps of different candidate adaptation ROIs where the probability value of all pixels in a certain adaptation ROI is less than a certain threshold.

First, given a certain frame F_t, we feed F_t into the segmentation model before test time adaptation and generate the temporary probability map M_f. Then, we generate 5 ROIs with 5 different candidate thresholds according to the value of M_f. The ROIs where the probability values of all pixels in this area are greater than α_{large} and α_{small}, receptively, are combined as the state of the RL model to choose t_p, as follows,

$$\begin{aligned} state_p = feature(\{i|i \in F_t, M_f(i) > \alpha_{large}\}) \\ +feature(\{i|i \in F_t, M_f(i) > \alpha_{small}\}). \end{aligned} \tag{7.5}$$

The ROIs where the probability values of all pixels in this area are greater than β_{large}, β_{medium} and β_{small}, receptively, are combined as the state of the RL model to choose t_n, as follows,

$$\begin{aligned} state_n = feature(\{i|i \in F_t, M_f(i) > \beta_{large}\}) \\ +feature(\{i|i \in F_t, M_f(i) > \beta_{medium}\}) \\ +feature(\{i|i \in F_t, M_f(i) > \beta_{small}\}). \end{aligned} \tag{7.6}$$

Note that β_{micro} is ignored in (7.6) to diminish the complexity of $state_n$, as $feature(\{i|i \in F_t, M_f(i) > \beta_{small}\})$ already provides sufficient information to help the RL model make the decision. We use the VGG model [96], pre-trained on the ImageNet classification dataset [23], to extract the features of these ROIs first. Then, we concatenate these features together as the state of the RL model. We use the first 5 convolutional blocks of the VGG19 model which results in a feature size of $\mathbb{R}^{7\times7\times512}$ for one ROI. For the RL model to choose t_p, the features of two ROIs will be concatenated to generate the final state $s_{t_p} \in \mathbb{R}^{7\times7\times1024}$. For the RL model to choose t_n, the features of three ROIs will be concatenated to generate the final state $s_{t_n} \in \mathbb{R}^{7\times7\times1536}$. Finally, states s_{t_p} and s_{t_n} will be fed into the corresponding RL model and result in the actions to choose the optimal thresholds t_p and t_n.

The reward function is defined as $r_t = g(s_t, a_p, a_n)$ which reflects the performance of the final segmentation result of each frame in the video sequence:

$$g(s_t, a_t, a_n) = \begin{cases} IOU + 1 & IOU > 0.1 \\ -1 & IOU <= 0.1 \end{cases}, \tag{7.7}$$

where IOU indicates the intersection-over-union (IOU) between the prediction and the ground-truth, which reflect the quality of the predicted segmentation.

7.2.4 Training in Actor-Critic Framework

In our work, we adopt the "actor-critic" framework [46] for RL training. In general, one "actor-critic" framework consists of two roles including an "actor" role to generate an action and a "critic" role to measure how good this action is. In this work, we need to select the optimal adaptation ROIs for both foreground and background separately. Therefore, we need two "actor-critic" model pairs, including one "actor-critic" pair for the foreground, and another pair for the background. Four individual RL models are deployed in total.

In our "actor-critic" framework, given a current frame F_t, the first step is to feed the state into the "actor" network and generate an action a, which is to choose the optimal adaptation ROIs. The corresponding reward r_t will also be obtained after conducting this action. r_t is decided by the IOU of the segmentation result according to (7.7).

In the training process, after the forward process, the "critic" network will be updated first in the valued-based way, as follows:

$$w = w' + \alpha * \delta_t \nabla_{w'} V_{w'}(s_t), \tag{7.8}$$

where

$$\delta_t = r_t + \gamma * V_{w'}(s_{t+1}) - V_{w'}(s_t). \tag{7.9}$$

In (7.8) and (7.9), w and w' indicate the weight of the "critic" model after and before update. α is the learning rate of the "critic" model. δ_t is the TD error which indicates the difference of the actual score and the predicted score. $V_{w'}(s_t)$ refers to the accumulated reward of state s_t which is predicted by the "critic" model before update. γ refers to the discount factor.

After the "critic" model has been updated, the "actor" model will be updated in a policy-based way, as follows:

$$\theta = \theta' + \beta * \nabla(log\pi_{\theta'}(s_t, a_t)) * A(s_t, a_t), \tag{7.10}$$

where $A(s, a)$ refers to the advantage function, and $A(s_t, a_t) = \delta_t$ according to (7.9), θ and θ' indicate the weight of the "actor" model after and before update. β is the learning rate of the "actor" model. Policy function $\pi(s, a)$ is a network whose input is the state s and a certain action a, and output is the probability of selection action a in state s.

In this way, when training the RL models, our "actor-critic" framework can avoid the shortage of value-based and policy-based methods. Instead of waiting until the end of the episode, our RL models can be updated at each step, which dramatically reduces the training time but maintains the RL training stability.

7.3 Implementation Details

7.3.1 Train the Segmentation Network

The proposed method trains the segmentation network following the strategy of [7] and [111]. The first step is to train an ImageNet pre-trained network on a pixel-level annotated dataset such as PASCAL VOC [26]. In the second step, we use the DAVIS2016 video dataset [83] to train the network so that the network is able to adapt to this dataset. We also fine-tune the network on the first frame of DAVIS test videos whose ground-truth annotation is provided. In this way, the segmentation network is well trained and can achieve a good result before test time adaptation.

7.3.2 Train the RL Model

Before training the RL model, the related data need to be stored in advance to accelerate the training process, which will be described in section 4.3. In terms of the training of the RL models, specifically, we divide all video sequences in DAVIS 2016 training set into video clips with a fixed number of frames. A video clip that includes 10/5 consecutive frames is used as a sample for the RL model to select foreground/background ROI. Using stored clips for training dramatically reduces the training time of the RL model.

We randomly select 20 clips as a batch for training the RL models. At the beginning of the training, the learning rate α for "actor" model is 1e-5, and the leaning rate β for "critic" model is 5e-5. The learning rate decreases gradually during the training, and it decreases by 1% for each 200 iterations. The discount rate γ for the reward is 0.9. In terms of values for candidate threshold t_p and t_n, we found α_{large}=0.97, α_{small}=0.7, β_{large}=0.4, β_{medium}=0.2, β_{small}=0.1 and β_{micro}= 0.01 works well through cross validation. The training of our RL models takes about 3 days on a NVIDIA GTX 1080 Ti GPU and a 12 Core Intel i7-8700K CPU@3.7GHz.

7.3.3 Accelerating RL Training

Segmentation with online adaptation is slow because the segmentation model should be updated for each frame. Meanwhile, RL training itself is also slow. Training the RL model heavily relies on a large number of attempts for different actions. Normally, to generate a well trained RL model, the model should be trained with more than one million iterations. If the running time for each training process is too long, the total time is unbearable. Thus, it is impossible to train the RL model in a regular way.

Method	DAVIS-16 J_m	DAVIS-16 F_m	SegTrack V2 J_m	Youtube Objs J_m	t(s)
PReMVOS [70]	84.9	**88.6**	-	-	30
OnAVOS [111]	<u>85.7</u>	84.8	66.7	77.4	13
CINM [4]	83.4	85.0	77.1	<u>78.4</u>	120
Lucid [42]	83.7	-	76.8	76.2	40
MSK [82]	79.7	75.4	72.1	75.6	12
OSVOS [7]	79.8	80.6	65.4	78.3	10
STV [118]	73.6	-	**78.1**	-	-
ObjFlow [108]	68.0	-	74.1	77.6	-
Ours	**87.1**	<u>86.1</u>	<u>77.5</u>	**79.5**	14

Table 7.1: Quantitative comparison with other methods on the DAVIS 2016, SegTrack V2 and Youtube-Object dataset. For J_m and F_m, the method wiht the best performance is bold, and the method with the second best performance is marked with underline.

To address this issue, inspired by the idea of sacrificing space to improve efficiency, we propose a novel multi-branch tree structure, as shown in Figure 7.4, to store all possible segmentation results using different adaptation ROIs into a repository in advance. Training an "actor-critic" framework needs 4 types of information including the action, the reward, and the states before and after the action. For each node in the multi-branch tree, a corresponding directory is generated. The image of the frame, the temporary probability map and the IOU of the segmentation result after executing a certain action are stored as files. All possible actions are stored as links to the next layer of nodes. In this way, the image and the probability map are used to generate the state. The IOU value is used to generate the reward. Finally, this repository will be organized in a multi-branch tree structure, whose stored data will be used to update the RL model.

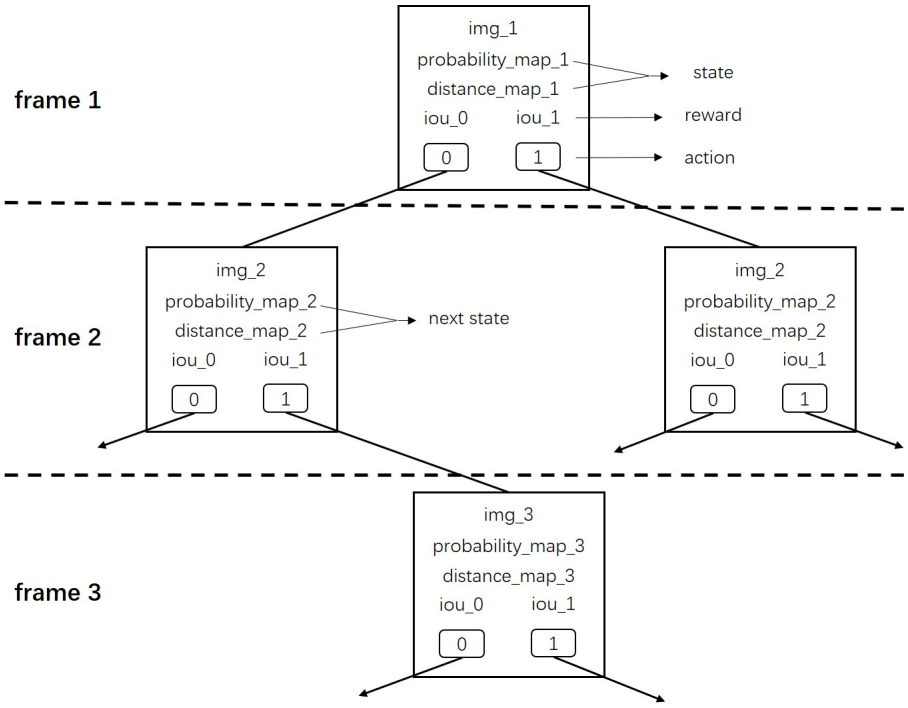

Figure 7.4: The data structure used to restore the related data to accelerate the RL model training. For RL model to select the t_n, distance map is required for the training, while the training for the RL mode to selected t_p does not need it.

7.4 Experiments

7.4.1 Experiment Setup

Our method is evaluated on three widely-used datasets including the DAVIS 2016 dataset [83], Youtube-Object data-set [86] and Segtrack V2 dataset [51]. DAVIS 2016 dataset consists of 50 high-quality video sequences and 3,455 frames, spanning multiple occurrences of common video object segmentation challenges such as occlusions, motion-blur and appearance changes. 30 video sequences of DAVIS 2016 are used for training, and 20 video sequences are used for testing. In DAVIS 2016, in each video sequence, only a single object instance is annotated. DAVIS 2017 dataset [85] extends the DAVIS 2016 dataset where multiple objects, rather than only one object, are annotated in each frame. As our method aims at single instance segmentation, we only do the experiment on DAVIS 2016. In Youtube-Object, there are 155 video sequences and a total of 570,000 frames. These video sequences are divided into 10 classes. The training set and testing set are not separated in Youtube-Object dataset so it is only used for testing. In SegTrack V2 dataset, there are 14 video sequences with more occlusion than appearance changes compared with Youtube-Object dataset.

We evaluate our method following the approach proposed in [83]. The adopted evaluation metrics include region similarity J and contour accuracy F. The region similarity is calculated as $J = \left|\frac{m \cap gt}{m \cup gt}\right|$ by the intersection-over-union between the predicted segmentation m and the ground-truth gt. The contour accuracy is defined as $F = \frac{2P_c R_c}{P_c + R_c}$, which indicates the trade-off between counter-based precision P_c and recall R_c.

7.4.2 Comparison with State-of-the-art

In this section, we will compare our proposed work with other state-of-the-art semi-supervised video object segmentation methods, including PReMVOS [70], OnAVOS [111], CINM [4], Lucid-Tracker [42], MSK [82], OSVOS [7], STV [118], ObjectFlow [108]. Note that OSVOS-S [72] is not included in the comparison list as it utilizes additional dataset for training.

Table 7.1 summarizes the quantitative results of recent methods on the DAVIS 2016 validation set consisting of 20 videos. The top 3 performing methods have been highlighted with different colors. It can be observed that our work has achieved outstanding results under both mean region similarity J_m and the mean contour accuracy F_m. Especially on mean region similarity J_m, our method achieves the best result which outperforms any existing state-of-the-art methods. Compared with the most competitive and related method OnAVOS [111], our method improves the

Method	DAVIS 2016
w/o adaptation	80.3 ± 0.4
foreground adaptation	82.1 ± 0.5
background adaptation	85.3 ± 0.5
full adaptation	86.5 ± 0.4
full adaptation + CRF	**87.1** ± 0.4
OnAVOS [111] *w/o* CRF	84.6 ± 0.6
OnAVOS [111]	85.7 ± 0.6
CINM [4]	84.2

Table 7.2: Ablation study on the contribution of individual RL model for the DAVIS 2016 dataset, measured by the mean region similarity J_m. WO indicates without.

mean region similarity J_m to 87.1%. It should be noted that the gain over [111] is solely due to the fact that better ROIs are obtained for online adaptation using RL. Note that, same to [75], according to the randomness of the segmentation network, the final accuracy may fluctuate around 0.4%. The proposed mean region similarity J_m is the average value from experiments of 10 times.

Figure 7.5 shows the qualitative segmentation masks for different methods. As can be observed, our method performs better on videos with significant appearance changes for the target object, for instance, the camel and breakdance video sequences. Especially when multiple similar objects are close to each other, *e.g.*, the camel sequence, our method has the ability to distinguish the target object from other similar objects successfully.

On SegTrack V2 and Youtube-Object datasets, as the training set and evaluation set are not split, all video sequences are used for evaluation. From Table 7.1, we can observe that our method also performs well on both datasets. Compared with OnAVOS [111], our method improves the mean region similarity J_m by 10.8% on SegTrack V2 dataset, which demonstrates the effectiveness of our RL models to choose the online adaptation ROIs. In addition, our approach also improves the mean region similarity J_m by 2.1% on Youtube-Object dataset. This result can show the robustness of our method on different evaluation datasets.

In addition, we compare our run time with other state-of-the-art methods, and the result is

Method	SegTrackV2
w/o adaptation	61.4 ± 0.6
foreground adaptation	66.2 ± 0.5
background adaptation	73.2 ± 0.5
full adaptation	76.6 ± 0.5
full adaptation + CRF	**77.5** ± 0.5
OnAVOS [111] *w/o* CRF	64.9 ± 0.6
OnAVOS [111]	66.7 ± 0.6
CINM [4]	77.1

Table 7.3: Ablation study on the contribution of individual RL model for the SegTrack V2 dataset, measured by the mean region similarity J_m. WO indicates without.

also reported in Table 7.1. Despite the fact that we improve the mean region similarity J_m of our baseline method with online adaptation [111] by 10.8% on SegTrack V2 dataset and 1.4% on DAVIS 2016 dataset, the run time of our method is only about 1 second longer than [111], which demonstrates the efficiency of our method.

7.4.3 Ablation studies

In this section, we conduct four ablation studies on our method using the testing video sequences of DAVIS 2016 dataset and SegTrack V2 dataset.

We conduct the first ablation studies on the DAVIS 2016 and SegTrack V2 datasets, where parts of our method are disabled to investigate the impact of each component. In this study, we also explore the contribution of each individual RL model in our method, one of these two RL models is disabled during this ablation study respectively, and the results will be compared with the result generated by the method with full RL models. Table 7.2 shows the result of this ablation study on the DAVIS 2016 dataset. On the DAVIS 2016 dataset, when using both foreground and background adaptation, we obtain the mean region similarity J_m of 86.5% without CRF [48]. J_m of our method with full adaptation is greater than the method without any adaptation by 6.2%, which demonstrates the effectiveness of our online adaptation approach. In addition, compared

Figure 7.5: Visualization of segmentation masks and the mean region similarity J_m for different methods on the DAVIS 2016 dataset.

with the rough approach to choose the adaptation ROI adopted by OnAVOS [111], before CRF, our method is also 1.9% greater than it. After executing CRF, the gain degrades a little to 1.4%, which demonstrates that our flexible way to choose the optimal adaptation ROI for each frame is significant for the final segmentation result. To study the individual influence of each RL model, we remove the RL model to choose the optimal adaptation ROI for background, obtaining the method **foreground adaptation**, and remove the model to choose the adaptation ROI for the foreground, obtaining the method **background adaptation**. As can be observed from the Table 7.2, using **foreground adaptation** method and **background adaptation** method obtains J_m of 82.1% and 85.3% on DAVIS 2016 dataset, respectively, which indicates that both RL models improve the segmentation result while the RL to choose the optimal adaptation ROI for background makes a larger contribution to the final segmentation result. This observation also explains the reason why we have more threshold candidates for background than foreground. We also conduct the same ablation studies on the Segtrack V2 dataset and obtain similar results as can be observed from Table 7.3. When using both foreground and background adaptation, we obtain J_m of 61.4% without CRF. J_m of our method with full adaptation is greater than the method without any adaptation by 15.2%. The comparison between the rough approach to choose the adaptation ROI adopted by OnAVOS [111] and the proposed method is conducted on Segtrack V2 dataset as well, before CRF. The result is that J_m of the proposed method is 11.7% greater than the baseline method, which is a dramatic improvement of the mean region similarity. After the process of CRF, the gain degrades a little to 10.8%, which is still a huge improvement. To study the individual influence of each RL model on Segtrack V2 dataset, we also adopt the method **foreground adaptation** and the method **background adaptation**, as on the DAVIS 2016 dataset. As can be observed from the Table 7.3, using **foreground adaptation** method and **background adaptation** method obtains 66.2% and 73.2% J_m on Segtrack V2 dataset, respectively, which also indicates that both of the two RL models contribute to the improvement of segmentation result. This experiment also demonstrates the importance of the optimal adaptation ROI mining.

The purpose of the second ablation experiment is to demonstrate that it is important to select different t_n and t_p for each specific frame, rather than adopting a particular fixed set of thresholds. In other words, no matter which set of t_n and t_p are chosen, as long as these thresholds are fixed for all frames, the final segmentation result will be worse than the result generated by adopting the optimal thresholds selected by the RL models for each specific frame. More specifically, the RL model to select the adaptation ROI for the background has 4 candidate thresholds including $\{0.4, 0.2, 0.1, 0.01\}$. The RL model to select the adaptation ROI for the foreground has 2 candidate thresholds including $\{0.97, 0.7\}$. In this way, there are totally 8 possible combinations of t_p and t_n,

t_n	t_p	J_m
0.4	0.97	61.9 ± 0.6
0.4	0.7	61.2 ± 0.6
0.2	0.97	64.2 ± 0.6
0.2	0.7	62.5 ± 0.6
0.1	0.97	73.7 ± 0.6
0.1	0.7	69.6 ± 0.6
0.01	0.97	83.6 ± 0.6
0.01	0.7	80.1 ± 0.6
Ours		**87.1** ± 0.4

Table 7.4: Performance comparison between heuristic adaptation ROIs selection and RL-based adaptation ROIs selection, conducted on the DAVIS 2016 dataset, measured by the mean region similarity J_m. t_n indicates the threshold to choose the adaptation ROI for background. t_p indicates the threshold to choose the adaptation ROI for foreground.

which are listed in Table 7.4. In this experiment, each combination of t_n and t_p is adopted as the fixed thresholds for the segmentation model adaptation and its corresponding result is evaluated and compared with the result generated by the proposed method. As can be observed from Table 7.4, among these combinations, the most "conservative" one, *i.e.* t_p=0.97 and t_n=0.01 performs best. Note that these values are different from the final result of OnAVOS[111] because OnAVOS selects the background adaptation ROI according to the distance, rather than the value of the probability map. The performance decreases dramatically when the value of the fixed threshold gets close to 0.5 because the error of segmentation will propagate quickly when adopting these "greedy" thresholds for all frames. The highest obtained J_m (83.6 %) of the method adopting fixed thresholds, however, is still much lower than the result generated by the proposed method (87.1 %). According to the observation of this experiment, it is obvious that the improvement does result from the RL models that choose the optimal t_n and t_p for each individual frame, rather than the contribution of a certain fixed combination of t_n and t_p.

The purpose of the third ablation experiment is to demonstrate that the proposed method is able to find a better adaptation ROI, compared with the baseline method. The metric to evaluate the selected adaptation ROI is the IOU between the adaptation ROI and the ground-truth. In this way, we first calculate the IOU between the adaptation ROI selected by the RL model and ground-truth. Then, we calculate the IOU between the adaptation ROI selected by the baseline method and the ground-truth. Finally, we compare these two IOUs and their contributions to the final segmentation result. As can be observed from Table 7.5, on average, the IOU of the proposed method is 40.9 % and 19.4 % higher than the baseline method, for background and foreground respectively. This result also coincides with the facts reported in Table 7.2 and Table 7.3 that background ROI adaptation using RL brings more performance gain than foreground ROI adaptation. Finally, as more correct pixels are adopted to update the segmentation model in the proposed method, the segmentation model is able to properly adjust itself to the change of the target in the video. It is also the reason why the proposed method can achieve improvement in the final segmentation result.

The fourth ablation experiment is to study the influence of different states on the RL model. The adopted state for the RL model also greatly affects the convergence difficulty of the training process as well as the final segmentation performance. We design three different states to study this factor. The first state is to feed the initial image (3 channels) and the temporary probability map (1 channel) into a train-from-scratch ConvNet without using pre-trained VGG model. This is because the input of the VGG model should be a 3-channel image. The second one is to feed the initial image (3 channels) concatenated with several mask channels to indicate different adaptation

Video	IOU-B-RL	IOU-B-Base	IOU-F-RL	IOU-F-Base	J_m-RL	J_m-Base	ΔJ_m
blackswan	99.4	44.6	91.0	74.6	95.4	95.4	0
bmx-trees	98.9	75.1	39.9	15.6	55.5	52.5	3.0
breakdance	97.4	55.9	74.9	55.0	80.2	68.4	11.8
camel	98.8	45.0	89.9	70.8	93.8	84.0	9.8
car-roundabout	99.4	41.8	95.2	85.4	97.1	97.1	0
car-shadow	99.7	60.1	92.7	79.2	96.1	96.0	0
cows	99.1	43.1	91.6	76.9	94.6	94.6	0
dance-twirl	98.1	59.3	83.3	63.2	87.3	84.6	2.7
dog	99.3	52.6	92.6	78.9	95.1	95.1	0
drift-chicane	99.6	72.1	83.0	55.6	89.1	87.2	1.8
drift-straight	99.1	64.4	90.1	75.8	92.7	91.3	1.3
goat	99.0	58.4	88.8	73.7	91.2	91.1	0
horsejump-high	98.9	61.8	80.0	52.4	87.3	86.8	0.5
kite-surf	98.3	72.4	57.2	28.0	66.7	66.7	0
libby	99.3	66.3	74.9	45.1	86.1	86.1	0
motocross-jump	93.9	41.2	87.5	70.7	90.4	86.4	4.0
paragliding	97.1	69.1	58.7	44.9	62.5	62.5	0
parkour	99.4	67.3	85.1	61.7	91.8	91.4	0.4
scooter-black	98.6	58.2	86.1	68.1	89.8	89.0	0.8
soapbox	98.3	46.1	86.0	65.4	90.1	86.2	3.8
mean	98.6	57.7	81.4	62.0	86.5	84.6	1.9

Table 7.5: The quality comparison between the selected ROIs of the proposed method and the baseline method, as well as their influences on the final segmentation results, conducted on the DAVIS 2016 dataset, measured by the mean region similarity J_m. In this table, IOU indicates the IOU between the adaptation ROI selected and the ground-truth. J_m indicates the mean region similarity. B refers to the background and F refers to the foreground. RL refers to the RL model and Base refers to the baseline method. Δ refers to the difference value between the RL model and the baseline method. All values in this table are calculated before CRF.

areas, similarly, without the feature extracting of VGG model. The third one is the proposed one where we generate a set of images with different masks to indicate different adaptation areas. The feature of each image is extracted by the VGG model. Then, the combined feature will be fed into the RL model. When using the first two types of states, after full training, the RL model still cannot choose the optimal adaptation area for each frame, which demonstrates these states are not suitable for this task. We believe the main reason is that the pre-trained VGG model is able to extract a better feature that contains more information of the original image. Also, the first two ways to indicate the different adaptation areas are not sufficiently explicit and discriminative.

Chapter 8

Conclusion

8.1 Summary

This thesis aims at tackling the multi-modal target localization and tracking task, where the target is localized or tracked according to the given target information. Such information can be provided in different modalities, including the segmentation mask, bounding box and referring sentence of the target, each of which has its own characteristic. In this thesis, the merits and shortcomings of each candidate modality (*i.e.*, text, box, mask) are analysed. Furthermore, we divide such a task into several sub-tasks, including text-referred target localization, text-referred target tracking, box-referred target tracking and mask-referred target tracking.

New frameworks are well designed to tackle these sub-tasks respectively. For text-referred target localization under supervised learning, we proposed a reinforcement learning based image-level shrinking network to iteratively cut off a certain region within the original image, with the ultimate remaining region as the result. For text-referred target localization under weakly learning, a discriminative triad based matching and reconstruction mechanism are proposed to convert the referring sentence into a unified format. For text-referred target tracking, we make the first attempt to select the start-frame, instead of the first frame. For box-referring target tracking, we propose a reinforcement learning based network to choose the matching standard (appearance-based or IoU-based) and whether update the target template after each inference. For mask-referred target tracking, the optimal adaptation areas for online adaptation are further explored.

8.2 Future Works

There are still some challenges left for future works. The main challenge is the lack of a unified framework to handle these sub-tasks simultaneously, making the exploration of these closely-linked tasks completely separated for a long time. The biggest reason is still the huge modality gap between different modalities, and the segmentation model is not yet compatible with both the visual and linguistic inputs.

Thus, we plan to design a unified multi-modality target localization and tracking framework. Specifically, we simultaneously decompose the image and sentence into a number of tokens with the same size, serving as the input vectors for the proposed unified multi-modal framework. Then, the 2D image is reshaped into a sequence of flattened patches, and each word in the referring sentence is viewed as a token. Thus, the tokens from both visual and linguistic information can be treated equally through a self-attention based localization and tracking model, making the proposed unified framework flexible to handle either the visual or linguistic input. Extensive experiments will be conducted to well support these future works.

In addition, it is still difficult for existing methods to highlight the significant information and discard the non-target or ambiguous one. As for an example, existing methods tend to roughly pass the entire frame image into the segmentation model, causing numerous resources wasted on the regions unrelated to the target object and limited attention paid to the meaningful target region. At the same time, the filtration for the referring sentence is also never considered in existing methods. Thus, we plan to design unified multi-modal filter networks for both the visual and linguistic input. And the optimal region or phrase of interest within the frame image and referring sentence can be selected respectively.

Publications

Conference:

1. Iterative Shrinking for Referring Expression Grounding Using Deep Reinforcement Learning. Sun M, Xiao J, Lim EG, *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2021, pp. 14060-14069.*

2. Fast template matching and update for video object tracking and segmentation. Sun M, Xiao J, Lim EG, Zhang B, Zhao Y, *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2020, pp. 10791-10799.*

3. Feature representation matters: End-to-end learning for reference-based image super-resolution. Xie Y, Xiao J, **Sun M**, Yao C, Huang K, *In European Conference on Computer Vision (ECCV) 2020, pp. 230-245.*

4. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. Zhang B, Xiao J, Wei Y, **Sun M**, Huang K, *In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) 2020, pp. 12765-12772.*

Journal:

1. Discriminative Triad Matching and Reconstruction for Weakly Referring Expression Grounding. Sun M, Xiao J, Lim EG, Liu S, Goulermas JY, *In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 43, no. 11, pp. 4189-4195, Nov. 2021.*

2. Starting Point Selection and Multiple-standard Matching for Video Object Segmentation with Language Annotation. Sun M, Xiao J, Lim EG, Zhao Y, *In IEEE Transactions on Multimedia (TMM), doi: 10.1109/TMM.2022.3159403 (early access)*

3. Cycle-free Weakly Referring Expression Grounding with Self-paced Learning. Sun M, Xiao J, Lim EG, Zhao Y, *In IEEE Transactions on Multimedia (TMM), doi: 10.1109/TMM.2021.3139467 (early access)*

4. Adaptive ROI generation for video object segmentation using reinforcement learning. Sun M, Xiao J, Lim EG, Xie Y, Feng J, *Pattern Recognition (PR), 1;106:107465, Oct. 2020.*

5. Transformer-Based Language-Person Search With Multiple Region Slicing. Li H, Xiao J, **Sun M**, Lim EG, Zhao Y, *In IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 3, pp. 1624-1633, Mar. 2022.*

6. Progressive sample mining and representation learning for one-shot person re-identification. Li H, Xiao J, **Sun M**, Lim EG, Zhao Y, *Pattern Recognition (PR), 1;110:107614, Feb. 2021.*

Bibliography

[1] Hazan Anayurt, Sezai Artun Ozyegin, Ulfet Cetin, Utku Aktas, and Sinan Kalkan. Searching for ambiguous objects in videos using relational referring expressions. *arXiv preprint arXiv:1908.01189*, 2019.

[2] Mohit Bajaj, Lanjun Wang, and Leonid Sigal. G3raphground: Graph-based language grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[3] Ankan Bansal, Yuting Zhang, and Rama Chellappa. Visual question answering on image sets. *European Conference on Computer Vision (ECCV)*, 2020.

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] 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.

[6] Adel Bibi, Matthias Mueller, and Bernard Ghanem. Target response adaptation for correlation filter tracking. In *European Conference on Computer Vision (ECCV)*, 2016.

[7] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[9] Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han. IMRAM: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Kan Chen, Jiyang Gao, and Ram Nevatia. Knowledge aided consistency for weakly supervised phrase grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[12] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] Tianlang Chen, Jiajun Deng, and Jiebo Luo. Adaptive offline quintuplet loss for image-text matching. *European Conference on Computer Vision (ECCV)*, 2020.

[15] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[17] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[18] Jongwon Choi, Hyung Jin Chang, Sangdoo Yun, Tobias Fischer, Yiannis Demiris, and Jin Young Choi. Attentional correlation filter network for adaptive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] Fanyi Xiao Yong Jae Lee Daniel Bolya, Chong Zhou. YOLACT: Real-time instance segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *International Conference on Language Resources and Evaluatio (LRCE)*, 2006.

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[25] Xingping Dong, Jianbing Shen, Wenguan Wang, Yu Liu, Ling Shao, and Fatih Porikli. Hyperparameter optimization for tracking with continuous deep q-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[26] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[27] Hongwei Ge, Zehang Yan, Kai Zhang, Mingde Zhao, and Liang Sun. Exploring overall contextual information for image captioning in human-like cognitive style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[28] Longteng Guo, Jing Liu, Xinxin Zhu, Peng Yao, Shichen Lu, and Hanqing Lu. Normalized and geometry-aware self-attention network for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[29] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. Contrastive learning for weakly supervised phrase grounding. *European Conference on Computer Vision (ECCV)*, 2020.

[30] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[31] Yamin Han, Peng Zhang, Wei Huang, Yufei Zha, Garth Douglas Cooper, and Yanning Zhang. Robust visual tracking based on adversarial unlabeled instance generation with label smoothing loss regularization. *Pattern Recognition*, 97:107027, 2020.

[32] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[35] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. 2017.

[36] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[37] Chen Huang, Simon Lucey, and Deva Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[38] Tommi Jaakkola, Satinder P Singh, and Michael I Jordan. Reinforcement learning algorithm for partially observable markov decision problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1995.

[39] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[40] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[41] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[42] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for object tracking. In *The DAVIS Challenge on Video Object Segmentation*, 2017.

[43] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision (ACCV)*, 2018.

[44] Junyeong Kim, Minuk Ma, Trung Pham, Kyungsu Kim, and Chang D Yoo. Modality shifting attention network for multi-modal video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

[47] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *European Conference on Computer Vision (ECCV)*, 2018.

[48] Krähenbühl, Philipp, and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[49] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[50] Dezhuang Li, Ruoqi Li, Lijun Wang, Yifan Wang, Jinqing Qi, Lu Zhang, Ting Liu, Qingquan Xu, and Huchuan Lu. You only infer once: Cross-modal meta-transfer for referring video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[51] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013.

[52] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[53] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *European Conference on Computer Vision (ECCV)*, 2020.

[54] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[55] Yuxi Li, Ning Xu, Jinlong Peng, John See, and Weiyao Lin. Delving into the cyclic mechanism in semi-supervised video object segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

[57] Yongqing Liang, Xin Li, Navid Jafari, and Qin Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[58] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[59] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[60] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[61] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[62] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[63] Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. Graph structured network for image-text matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[64] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[65] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[66] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[67] Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Dechao Meng, and Qingming Huang. Adaptive reconstruction network for weakly supervised referring expression grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[68] Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Li Su, and Qingming Huang. Knowledge-guided pairwise reconstruction network for weakly supervised referring expression grounding. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2019.

[69] Ying Liu, Konstantinos Tountas, Dimitris A Pados, Stella N Batalama, and Michael J Medley. L1-subspace tracking for streaming data. *Pattern Recognition*, 97:106992, 2020.

[70] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision (ACCV)*, 2018.

[71] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[72] 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. *arXiv preprint arXiv:1709.06031*, 2017.

[73] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[74] Yunyao Mao, Ning Wang, Wengang Zhou, and Houqiang Li. Joint inductive and transductive learning for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[75] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[76] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[77] 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 *International Conference on Machine Learning (ICML)*, 2016.

[78] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[79] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[80] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[81] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[82] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[83] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[84] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *European Conference on Computer Vision (ECCV)*, 2018.

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

[86] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

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

[88] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[89] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[90] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016.

[91] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[92] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[93] Fawaz Sammani and Luke Melas-Kyriazi. Show, edit and tell: A framework for editing image captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[94] Enver Sangineto, Moin Nabi, Dubravko Culibrk, and Nicu Sebe. Self paced deep learning for weakly supervised object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):712–725, 2018.

[95] Hongje Seong, Seoung Wug Oh, Joon-Young Lee, Seongwon Lee, Suhyeon Lee, and Euntai Kim. Hierarchical memory matching network for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12889–12898, 2021.

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

[97] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[98] Mingjie Sun, Jimin Xiao, and Eng Gee Lim. Iterative shrinking for referring expression grounding using deep reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[99] Mingjie Sun, Jimin Xiao, Eng Gee Lim, Si Liu, and John Y Goulermas. Discriminative triad matching and reconstruction for weakly referring expression grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4189–4195, 2021.

[100] Mingjie Sun, Jimin Xiao, Eng Gee Lim, Yanchun Xie, and Jiashi Feng. Adaptive ROI generation for video object segmentation using reinforcement learning. *Pattern Recognition*, 106:107465, 2020.

[101] Mingjie Sun, Jimin Xiao, Eng Gee Lim, Bingfeng Zhang, and Yao Zhao. Fast template matching and update for video object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[102] Mingjie Sun, Jimin Xiao, Eng GEE Lim, and Yao Zhao. Cycle-free weakly referring expression grounding with self-paced learning. *IEEE Transactions on Multimedia*, 2021.

[103] Mingjie Sun, Jimin Xiao, Eng GEE Lim, and Yao Zhao. Starting point selection and multiple-standard matching for video object segmentation with language annotation. *IEEE Transactions on Multimedia*, 2022.

[104] Peng Sun, Peiwen Lin, Guangliang Cheng, Jianping Shi, Jiawan Zhang, and Xi Li. Ovsnet: Towards one-pass real-time video object segmentation. *arXiv preprint arXiv:1905.10064*, 2019.

[105] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

[106] Ruixue Tang, Chao Ma, Wei Emma Zhang, Qi Wu, and Xiaokang Yang. Semantic equivalent adversarial data augmentation for visual question answering. *European Conference on Computer Vision (ECCV)*, 2020.

[107] Alasdair Tran, Alexander Mathews, and Lexing Xie. Transform and tell: Entity-aware news image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[108] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[109] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[110] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[111] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In *The 2017 DAVIS Challenge on Video Object Segmentation Workshops (CVPR)*, 2017.

[112] Paul Voigtlaender, Jonathon Luiten, and Bastian Leibe. Boltvos: Box-level tracking for video object segmentation. *arXiv preprint arXiv:1904.04552*, 2019.

[113] Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lin Ma. Consensus-aware visual-semantic embedding for image-text matching. *European Conference on Computer Vision (ECCV)*, 2020.

[114] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[115] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[116] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood Watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[117] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[118] Wenguan Wang, Jianbing Shen, Jianwen Xie, and Fatih Porikli. Super-trajectory for video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[119] Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[120] Hefeng Wu, Yafei Hu, Keze Wang, Hanhui Li, Lin Nie, and Hui Cheng. Instance-aware representation learning and association for online multi-person tracking. *Pattern Recognition*, 94:25–34, 2019.

[121] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[122] Yanchun Xie, Jimin Xiao, Kaizhu Huang, Jeyarajan Thiyagalingam, and Yao Zhao. Correlation filter selection for visual tracking using reinforcement learning. *arXiv preprint arXiv:1811.03196*, 2018.

[123] Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*, 2019.

[124] 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 *International Conference on Machine Learning (ICML)*, 2015.

[125] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[126] Sibei Yang, Guanbin Li, and Yizhou Yu. Cross-modal relationship inference for grounding referring expressions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[127] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[128] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. *European Conference on Computer Vision (ECCV)*, 2020.

[129] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[130] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[131] Raymond Yeh, Jinjun Xiong, Wen-Mei Hwu, Minh Do, and Alexander Schwing. Interpretable and globally optimal prediction for textual grounding using image concepts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[132] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[133] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision (ECCV)*, 2016.

[134] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[135] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[136] Dingwen Zhang, Deyu Meng, and Junwei Han. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):865–878, 2016.

[137] Lu Zhang, Zhe Lin, Jianming Zhang, Huchuan Lu, and You He. Fast video object segmentation via dynamic targeting network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[138] Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z Li. Context-aware attention network for image-text retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[139] Yuting Zhang, Luyao Yuan, Yijie Guo, Zhiyuan He, I-An Huang, and Honglak Lee. Discriminative bimodal networks for visual localization and detection with natural language queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[140] Yuanen Zhou, Meng Wang, Daqing Liu, Zhenzhen Hu, and Hanwang Zhang. More grounded image captioning by distilling image-text matching model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[141] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton Van Den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[142] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[143] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision (ECCV)*, 2014.