Grounding-Tracking-Integration

Zhengyuan Yang, *Graduate Student Member, IEEE*, Tushar Kumar, Tianlang Chen, *Graduate Student Member, IEEE*, Jingsong Su, and Jiebo Luo, *Fellow, IEEE*

***Abstract*— In this paper, we study tracking by language that localizes the target box sequence in a video based on a language query. We propose a framework called GTI that decomposes the problem into three sub-tasks: Grounding, Tracking, and Integration. The three sub-task modules operate simultaneously and predict the box sequence frame-by-frame. "Grounding" predicts the referred region directly from the language query. "Tracking" localizes the target based on the history of the grounded regions in previous frames. "Integration" generates final predictions by synergistically combining grounding and tracking. With the "integration" task as the key, we explore how to indicate the quality of the grounded regions in each frame and achieve the desired mutually beneficial combination. To this end, we propose an "RT-integration" method that defines and predicts two scores to guide the integration: 1) R-score represents the Region correctness whether the grounding prediction accurately covers the target, and 2) T-score represents the Template quality whether the region provides informative visual cues to improve tracking in future frames. We present our real-time GTI implementation with the proposed RT-integration, and benchmark the framework on LaSOT and Lingual OTB99 with highly promising results. Moreover, we produce a disambiguated version of LaSOT queries to facilitate future tracking by language studies.**

***Index Terms*— Tracking by language, visual grounding, vision+language, object tracking.**

I. Introduction

GIVEN a video and a language query, tracking by language [1] is the task of predicting the box sequence of the referred object based on the input language query, as shown in Figure 1 (a). The grounded box sequences are predicted sequentially in each frame of the input video. Compared to specifying the target by drawing a box as in object tracking [2]–[5], providing a language query is a natural way of human-computer interaction. The language specification provides the clear semantic meaning of the target and thus alleviates certain failures in object tracking caused by appearance changes, occlusion, box drifting, *etc*. Tracking by language also opens up applications such as starting at an arbitrary time-step and searching in a video corpus in parallel. In addition, good tracking by language model benefits various related research problems, such as language-based video retrieval [6] and video QA [7].

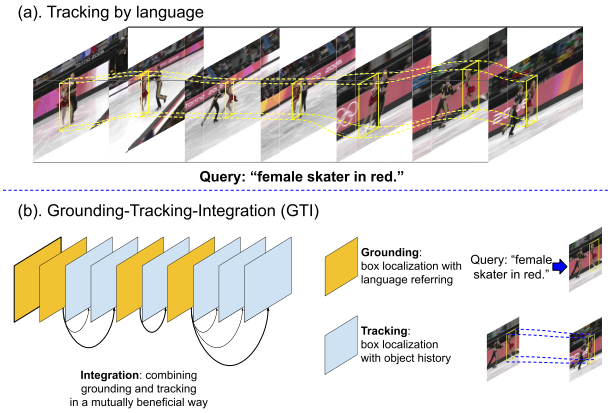

Fig. 1. *Tracking by language* aims to localize the tubelet specified by a language query. We propose a GTI framework that decomposes the problem into three sub-tasks: grounding, tracking, integration. This study focuses on the key "integration" task.

Naturally, two kinds of information are available in tracking by language. On the one hand, the language query contains target specifications in all frames. On the other hand, the history of the grounded image patches in previous frames provides cues for the target. Therefore, tracking by language can be approached either from language referring ("grounding") or visual patch matching ("tracking") perspectives. For the first perspective, "grounding" approaches the problem by processing each frame independently. However, "grounding" methods frequently fail in frames of degraded visual qualities. The grounded regions also tend to be inconsistent throughout time, as "grounding" alone exploits no neighboring frame similarities in videos. For the second perspective, "tracking" localizes the region based on a given box in previous frames. When initialized with an ideal given box (by grounding), "tracking" generally provides tubelets of better qualities than "grounding". However, 'tracking" suffers from bad initialization when the language grounded region either refers to the incorrect object or does not contain informative visual cues of the target for tracking.

This study builds on the understanding that neither "grounding" nor "tracking" alone solves the tracking by language problem, while the combination can compensate for each other's weaknesses. "Tracking" has the potential to correct

Manuscript received July 5, 2020; revised October 5, 2020; accepted November 4, 2020. Date of publication November 17, 2020; date of current version September 3, 2021. This work was supported in part by the National Science Foundation (NSF) under Award IIS-1704337, Award IIS-1722847, and Award IIS-1813709, Twitch Fellowship, as well as our corporate sponsors. This article was recommended by Associate Editor J. Wang. *(Corresponding author: Jiebo Luo.)*

Zhengyuan Yang, Tushar Kumar, Tianlang Chen, and Jiebo Luo are with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA (e-mail: zyang39@cs.rochester.edu; tusharku@cs.rochester.edu; tchen45@cs.rochester.edu; jluo@cs.rochester.edu).

Jingsong Su is with the School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: jssu@xmu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2020.3038720.

Digital Object Identifier 10.1109/TCSVT.2020.3038720

"grounding" failures based on the information from adjacent frames, whereas "grounding" could improve "tracking" by re-initializing the tracker with better language grounded regions.

In this study, we propose a **GTI** framework, where we decompose the tracking by language task into three sub-tasks: **G**rounding, **T**racking and **I**ntegration. Given a frame, "grounding" localizes the region directly from the input language query. "Tracking" predicts by using the history of grounded regions as tracking templates, *i.e*., the "tracking" predicted region should be visually similar to the region in tracking templates. "Integration" combines the two perspectives in a mutually beneficial way to obtain better final predictions. As shown in Figure 1 (b), "integration" selects whether "grounding" or "tracking" is more important in each frame, and generates the final box prediction accordingly. In frames where "grounding" is assigned higher importance, the language grounded region is included in the region history to help "tracking" in future frames. The three modules function simultaneously to generate tubelet predictions frame-by-frame.

A. Criteria for Integration

While a wide range of "grounding" [8]–[11] and "tracking" [12]–[14] methods exist, the "integration" problem is unique in tracking by language and we are not aware of any proper method that can be directly applied. "Integration" with pre-defined rules or fixed weights in all frames [1] generally shows limited performance. Because such naive methods operate independently of the per-frame context and grounded regions, they neither manage to correct grounding failures with tracking results nor strengthen future tracking with selected grounded regions. Instead, the "integration" module should operate adaptively in each frame by referencing the corresponding visual input, language query, and grounded region. To be specific, a good "integration" module should satisfy the following criteria: 1) The module should predict if the grounded region accurately covers the target, and assign higher importance to "grounding" in such frames. 2) The module should predict if the grounded region contains informative visual cues of the target that could improve the tracker, and include such region into the object history. 3) The module should be light-weighted and fast.

B. Mechanism for Integration

We propose a new paradigm for the "integration" problem named *RT-integration*. In each frame, we predict two scores to guide the "integration". R-score reflects the **R**egion correctness, *i.e*., whether the grounded region accurately covers the language referred object. T-score reflects the **T**emplate quality, *i.e*., whether the grounded region contains discriminative visual cues to help "tracking". High RT-scores indicate the high importance of "grounding". In such frames, we take "grounding" predictions both as the outputs and future tracking templates, whereas in the remaining frames, the "tracking" prediction is adopted as the outputs to correct possible grounding failures. We derive the ground-truth RT-scores from box annotations and train a separate module for RT-score prediction.

Finally, we present our real-time implementation of the GTI framework with the proposed RT-integration. We benchmark the proposed framework on LaSOT [15] and Lingual OTB99 [1] with highly promising results. As the original language queries in LaSOT can be ambiguous [15], we clean the dataset by replacing the ambiguous queries with new annotations. Our contributions are:

- We propose a Grounding-Tracking-Integration (GTI) framework for tracking by language.
- We propose "RT-integration" that adaptively integrates grounding and tracking with the region correctness score and template quality score predicted in each frame.
- Our real-time implementation of the GTI framework shows highly promising results on multiple datasets.
- We clean up the ambiguous queries in LaSOT [15] to facilitate future tracking by language studies.

II. Related Work

A. Tracking With Box Specifications

Tracking returns the tubelet of the specified object in a video. Our study is related to the traditional object tracking [2]–[5], where the ground-truth box in the first frame (the tracking template) specifies the object of interest. Correlation filter based methods [16]–[18] show good efficiency and accuracy on the task. Recently, the Siamese network based trackers [13], [14], [19] also show promising performance. In this study, we study the problem of using language queries to replace boxes as the target specification.

B. Tracking With Language Cues

Several previous studies explore tracking with language cues [1], [15], [20], [21]. Wang *et al.* [21] adopt language queries as the extra information alongside with boxes for tracking. LaSOT [15] is a recently proposed large scale tracking dataset that has auxiliary language query annotations. Li *et al.* [1] first introduce the tracking by language task and propose a Lingual Specification Attention Network (LSAN). The authors encode the region history and language query as the parameters for two independent dynamic filters, and generate per-frame tracking and grounding predictions accordingly. The predictions are then fused with a fixed weight in all frames. We later show that LSAN is a special case of the GTI framework with a naive integration module. Feng *et al*. [22] propose to solve tracking by language with the tracking by detection approach, and the tracking prediction is fed into the detectors to refine the detection results.

C. Visual Grounding

Visual grounding [8], [11], [23] is the task of localizing the referred region in an image given a language query. Most previous methods [10], [24]–[27] follow a two-stage pipeline, where a number of region candidates are first detected, followed by a language-based ranking stage to find the most relevant region. The recently proposed one-stage methods [9], [28], [29] conduct visual-textual fusion at image level and improve both the accuracy and inference speed. Recent studies [30], [31] explore the visual grounding problem in videos.

D. Self-Evaluation Scores

Our proposed "RT-integration" module is related to previous studies [32]–[35] on learning self-evaluation scores. In object tracking, ATOM [32] predicts the Intersection over Union (IoU) between the tracking output and the ground-truth target by taking tracking templates as references. Our method is more relevant to IoU prediction in object detection [33] and instance segmentation [35], where no template references are available. IoU-Net [33] proposes an IoU prediction module on top of the detection backbone [36] to predict the localization confidence. MS R-CNN [35] extends the idea for instance segmentation.

Previous studies in video object detection [37], [38] explore the similar idea of score-based integration. Tang *et al.* [38] propose to generate accurate and reliable object tubelet prediction by linking short tubelets based on the temporal overlap. Kang *et al.* [37] show better video object detection can be achieved by combining the confidence score of a per-frame object detector and a tracker. Going beyond self-evaluation score prediction based on the objectiveness, the "integration" task in tracking by language poses extra requirements of 1) predicting visual-textual similarity, 2) predicting template quality, and 3) being fast.

III. Grounding-Tracking-Integration

Given a natural language query for a video, we hope to return the box sequence of the referred object. Different from object tracking [13], [14], the references are specified by a language query instead of the ground-truth bounding box in the first frame. We propose a Grounding-Tracking-Integration (GTI) framework to approach the problem. As shown in Figure 2, the three modules operate simultaneously and generate box predictions frame-by-frame. In each frame, "grounding" takes the frame and language query as input for object localization. "Grounding" operates independently in each frame and does not accumulate errors. However, it may fail due to the errors of grounding methods. "Tracking" predicts the box based on the history of language grounded regions. When provided a correct region in nearby frames as the template, "tracking" generally generates better box predictions than "grounding." However, "tracking" often accumulates the error from the given template, and decreases in performance when the temporal distance between the template and current frame increases. "Integration" looks at both grounding and tracking predictions, and generates the final prediction.

IV. RT-Integration

We investigate the "integration" task in the GTI framework. The goal of this sub-task is to combine "grounding" and "tracking" in a mutually beneficial and overall synergistic way to generate better final predictions. In frames where "grounding" predictions are of good quality, including such grounded regions into tracking templates strengthens the tracker for future frames. In frames where "grounding" is likely to fail, adopting the "tracking" prediction generally leads to better final predictions. To achieve such a mutually beneficial combination, "integration" should predict when "grounding"

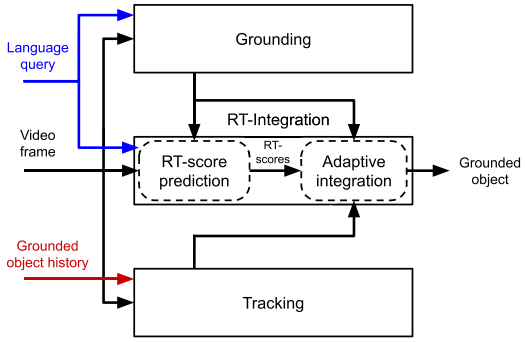

Fig. 2. The block diagram of the GTI framework with our proposed RT-integration.

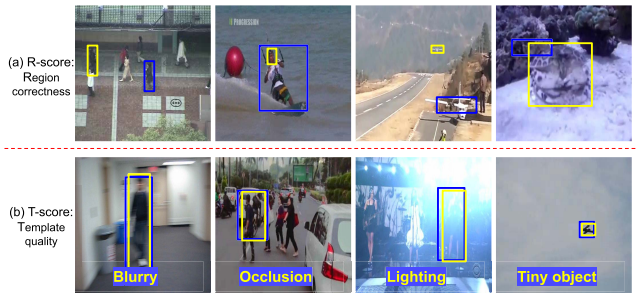

Fig. 3. Example frames with low region correctness scores (top row) and low template quality scores (bottom row). Blue/ yellow boxes are grounding predictions [9]/ ground-truth, respectively.

is of good quality or likely to fail, and adjust the grounding-tracking importance in each frame accordingly.

The core idea of our proposed RT-integration is to represent the grounding and tracking importance in each frame as two scores, namely the RT-scores, where higher score values indicate the better quality and thus higher importance of "grounding." The R-score reflects if the grounded region precisely covers the target, and the T-score shows if the grounded region contains visual cues that can improve the tracker. In frames with high RT-scores, the "grounding" prediction is selected as the output and used to update the tracker, whereas the remaining frames are processed by "tracking." This study focuses on how to properly define and precisely predict the RT-scores.

A separate module is trained in a fully supervised way to predict the RT-scores, as shown in Figure 2. The input to the module is the visual-textual feature and the language grounded region in a frame. The output is the corresponding RT-score prediction, as shown in Figure 4. During training, the ground-truth RT-scores are derived from the box annotations and are used to train the module. Essentially, "integration" can be regarded as a self-judge process for the framework to examine whether the language grounded region in a frame is valid as the output and new template. Section IV-A introduces the definition of the derived RT-scores. Section IV-B presents the details of the module architecture and training procedure. During inference, RT-scores are predicted with the trained module in each frame, and guide the adaptive integration that synergistically combines grounding and tracking to generate final predictions. Section IV-C introduces the RT-score-guided

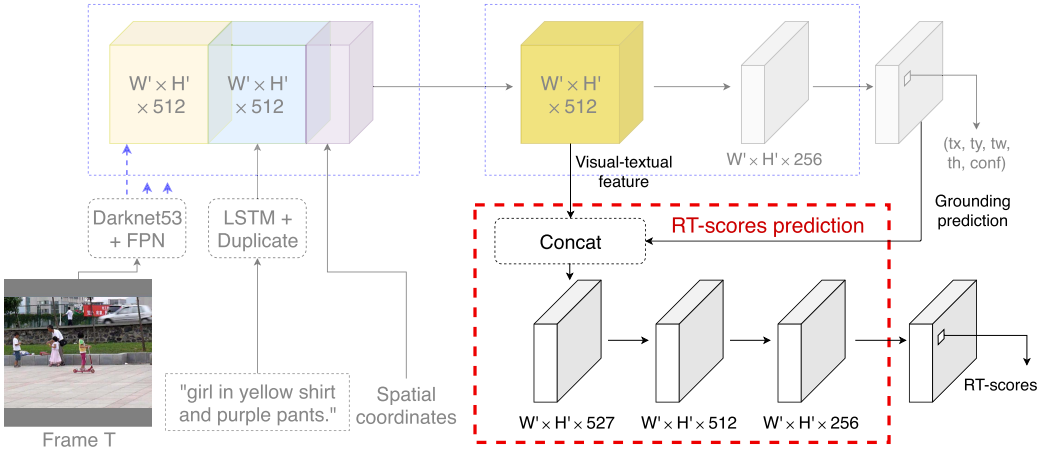

Fig. 4. The network architecture for RT-scores prediction. The backbone grounding method [9] is shown in translucent colors outside the red box. Feature pyramid heads are used in the grounding method. For visualization purpose, we only show one head.

adaptive integration in each frame. A complete inference pipeline of the GTI framework is presented in Section V.

A. RT-Scores

Two factors are essential for an ideal "integration." First, "integration" should predict if the language grounded region accurately covers the target. The state-of-the-art grounding method [9] commonly fails in frames with multiple objects of the same kind, tiny targets, and limited visual qualities (*e.g.*, the top row in Figure 3), when "tracking" should be adopted to correct the errors. Second, "integration" should predict if the grounded region contains visual cues that can strengthen the tracker to help future frames. The bottom row in Figure 3 shows negative examples with limited frame qualities or improper target statuses.

We propose to model the two factors with two scores respectively, namely the RT-scores. The R-score (**R**egion correctness) models how accurately the grounded box covers the target. In frames with low R-scores, grounding is likely to be failed and can be corrected by tracking. We define the R-score as the Intersection over Union (IoU) between the language grounded region and the ground-truth box. We collect the per-frame visual-textual feature and grounded region pairs with a visual grounding method [9] and calculate the R-score in each sample accordingly. The T-score (**T**emplate quality) models how well the target image patch in a frame serves as the tracking template, *i.e.*, if purely relying on the visual similarity between the target image patch (the tracking template) and the candidates in future frames, how accurate the localization results will be. In frames with high T-scores, the grounding predictions contain informative visual cues that could improve tracking, while some patches have low T-scores and do not benefit tracking (as shown in Figure 3 (b)). In our study, we obtain the ground-truth T-score by conducting tracking with a fixed tracker [13]. To be specific, we initialize the tracker with the ground-truth target region in a given frame, and conduct tracking in all remaining frames. With the fixed tracker and the almost identical tracking video (except the given template frame itself), only the template patch quality influences the tracking performance. Therefore, the obtained mean IoU reflects the desired template quality and is adopted as the ground-truth T-score.

B. Score Prediction

We next introduce the proposed module for RT-scores prediction. In each frame, the module refers to the frame, query, and grounded box to generate the RT-score prediction. We re-use the fused feature from "grounding" as the per-frame visual-textual representation to boost the inference speed. As shown in Figure 4, the proposed module takes the grounded region and the fused visual-textual feature from "grounding" [9] as inputs and predicts the scores for the grounded region. The module consists of three stand-alone 1×1 convolutional layers. The RT-scores in the same spatial location as the top-1 "grounding" prediction is output as the final score prediction.

The score prediction module is trained separately from "grounding" and "tracking." We model the R- and T-score predictions as two separate regression problems trained by the smoothed-L1 loss [39]. With a pre-trained grounding model [9], we generate training samples by collecting the triplets of visual-textual features, grounded regions, and derived RT-scores. During training, we filter out the samples with a grounding confidence score of less than 0.5. Such grounded regions are likely to be incorrect and can be well identified by grounding confidences. We find the filtering simplifies the score prediction problem and empirically leads to better performances. During inference, we consider such a region incorrect and directly set the R-score to 0.

C. Adaptive Integration

In each frame, adaptive integration updates the tracking template and generates the final prediction based on the per-frame scores, instead of pre-defined rules or fixed weights. With the RT-scores predicted, there exist multiple ways of generating the final prediction and updating the template based on the score, *e.g.*, score-guided soft weighted fusion, or hard switching between grounding and tracking. We observe that the quality of the score prediction instead of the exact integration method influences the performance the most. Therefore,

Algorithm 1 Our Implementation of GTI

Input: Video $\mathcal{V} = \{v_1, \ldots, v_n\}$ and Query Q
Function $\mathfrak{G}$ is the "grounding" module.
Function $\mathfrak{T}$ is the "tracking" module.
Function $\mathfrak{I}$ is the RT-score prediction module.
S is the saved score for the current template T.
b_g is the per-frame grounding prediction.
s_t is the RT-scores for the grounding prediction.
λ is the decay rate of the saved score S.
Output: Per-frame object boxes $\mathcal{B} = \{b_1, \ldots, b_n\}$

```
b_g ← G(v_1, Q)                          // Grounding
s_1 ← I(v_1, Q, b_g)                     // Initial RT-scores
b_1, S, T ← b_g, s_1, b_g                // Output, init tracker
for t in 2, ..., n do
    b_g ← G(v_t, Q)
    s_t ← I(v_t, Q, b_g)                 // Predicted RT-scores
    /* If grounding is more important */
    if S < s_t then
        b_t, S, T ← b_g, s_t, b_g
    /* If tracking is more important */
    else
        b_t ← T(v_t, T)
        S ← S * λ
    end
end
```

we present a vanilla version of hard switching as follows, and defer the introduction and experiments of the alternatives to Section VI-G. First, the R- and T-scores are multiplied in each frame to obtain a combined score that guides "integration." We consider "grounding" more important whenever the predicted combined score is higher than the previously saved highest value. In such frames, we adopt "grounding" as the output and update the template. Otherwise, we output tracking predictions.

With the same set of importance scores, we observe that the exact score-guided adaptive integration method, *e.g.*, soft weighted fusion or hard switching, has no significant influence on the final performance. Instead, for "integration," we find it important to accurately define and predict the importance scores based on the per-frame context. We detail the analyses and experiments in Section VI-G.

V. Implementation of GTI

In this section, we present our real-time implementation of the GTI framework. We introduce the adopted "grounding" and "tracking" modules, as well as the overall pipeline.

A. Grounding

Given a frame, the "grounding" module predicts a region based on the language query. We adopt the one-stage visual grounding [9] as the grounding module because of its state-of-the-art accuracy and real-time inference speed. The grounding method merges language and spatial features into YOLOv3 [40] for visual grounding. DarkNet-53 [40] and feature pyramid network [36] are used to encode the visual feature. With an input resolution of 256×256, the three feature pyramid heads have the spatial resolutions of 8×8, 16×16 and 32×32, respectively. Similar to one-stage object detection [40], the grounding method outputs multiple box predictions at each of the $8 \times 8 + 16 \times 16 + 32 \times 32 = 1344$ locations. With three anchor boxes predicted at each location, the method outputs $3 \times 1344 = 4032$ grounding predictions per frame. Each predicted region consists of five values, i.e. the relative position, width, height and the confidence score. The prediction with the highest confidence score is output as the final grounded region in each frame.

B. Tracking

Given a frame, the "tracking" module localizes the target based on the language grounded region history in previous frames. We adopt the SiamRPN++ [13] as the tracker in our implementation while various other object tracking methods [32], [41] can also be directly applied, as shown in ablation studies. SiamRPN++ is a Siamese network based tracker that models tracking as the feature cross-correlation between the tracking template and current frame.

C. Inference

We then present the inference pipeline on a testing video in Algorithm 1. Given no region history is available in the first frame, the "grounding" result is directly adopted as the output and used to initialize "tracking." The predicted RT-scores are also saved. In all the following frames, the three modules operate simultaneously. "Integration" predicts the RT-scores in a frame and compare it to the saved value. Whenever a higher score appears, we adopt "grounding" as the output, and update tracking template T and saved score S accordingly. In remaining frames, "tracking" is adopted as the output.

VI. Experiments

A. Datasets

1) Disambiguated LaSOT: LaSOT [15] contains 1,400 videos with auxiliary language queries. We follow the split [15] that uses 1,120 videos for training and 280 videos for testing. The averaged video length is around 2,500 frames.

The original LaSOT dataset [15] contains auxiliary language queries that might provide ambiguous target specifications. For example, in Figure 5 (a), the referred glass can not be distinguished based on the original query. To facilitate tracking by language studies, we clean the LaSOT queries by replacing the ambiguous queries with new annotations. As the first step, annotators are presented with the video, target tubelet, and the original language query in LaSOT, and are asked to label if the target can be distinguished based on the original query. The collected annotations show that 322 out of the 1,400 original video queries are ambiguous. Annotators then generate new queries that have clear target specifications. Extra descriptions of the target's location, color, size, relationships are included in the cleaned queries. In the end, we verify the quality of the generated queries. Among the 322 updated queries, 80 queries are still ambiguous, *i.e.*, at least one out of two annotators can not distinguish the target based on the

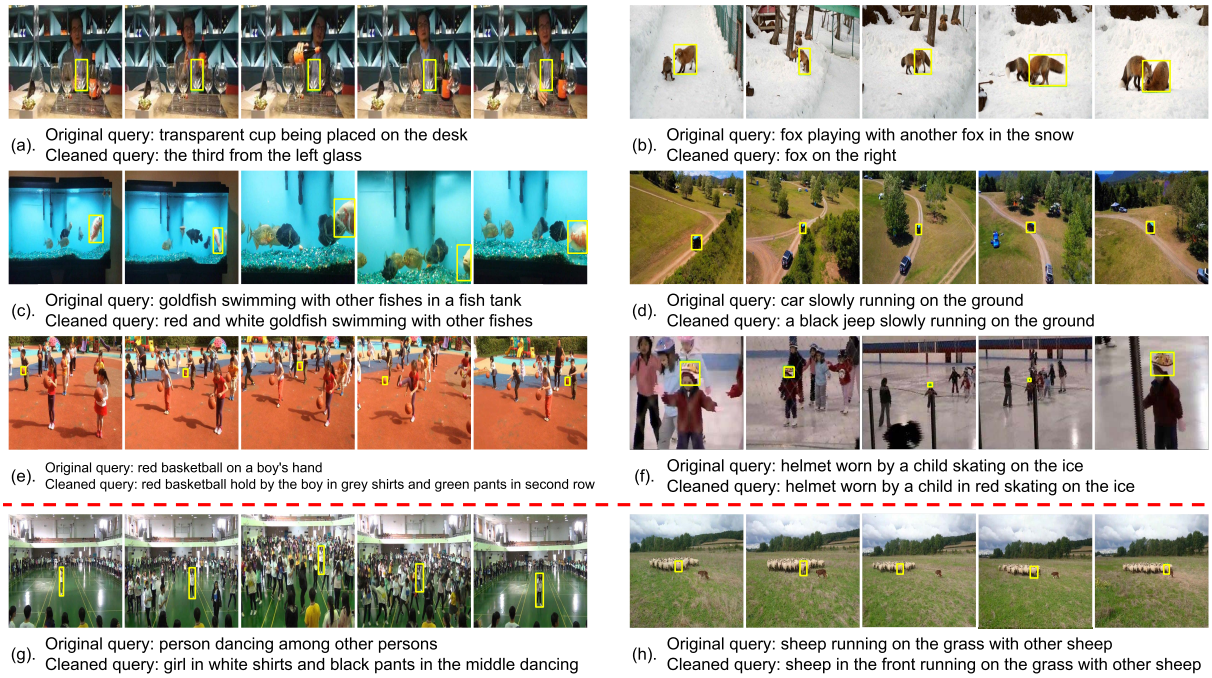

Fig. 5. Examples of the disambiguated queries in LaSOT. The first three rows show the disambiguated queries, and the last row presents the samples that annotators find difficult to refer by language.

new query. We fail to generate precise queries for all targets because some videos contain visually identical objects and are not proper for tracking by language studies (*e.g.*, Figures 5 (g) and (h)).

We provide representative examples of the updated queries in Figure 5. Figures 5 (a) and (b) add extra location descriptions to disambiguate the query. Figures 5 (c) and (d) include color and entity descriptions to provide the target specification. Figures 5 (e) and (f) provide relationships and other detailed descriptions to generate a precise target specification. After the manual annotation, a small portion of samples is still ambiguous because the language query alone can not generate a clear specification for the given target. For example, in Figures 5 (g) and (h), visually similar objects exist and make language referring difficult.

2) Lingual OTB99: Lingual OTB99 [1] augments the OTB100 object tracking dataset [3], [42] with natural language descriptions. One query is annotated per target object. We follow the training/ testing split [1] that uses the OTB51 videos for training and the remaining 48 videos for testing. The averaged video length is around 600 frames.

3) Lingual ImageNet Videos: The Lingual ImageNet videos dataset [1] augments the ImageNet Video Object Detection dataset [43] with one query per target object. We follow the same split [1] that uses 50 videos for training and the other 50 for testing. The averaged video length is around 270 frames.

The targets and videos in the Lingual ImageNet videos dataset [1] used in previous studies [1] are far from real and oversimplify the problem, and thus are not suitable for study. We show the analyses in Section VI-D.

B. Implementation Details

1) Evaluation Criteria: We evaluate the methods with precision and success scores [4]. The precision score reflects the percentage of frames where the estimated location falls within a given threshold of 20 pixels with the target. The success plot shows the ratio of success frames under an IoU threshold ranging from 0 to 1. The Area Under Curve (AUC) of the success plot represents the averaged success rates with different sampled thresholds and is used for evaluation. We follow the online tracking setting that the method only observes the previous and current frames for prediction.

2) Training Details: We train the score prediction module in RT-integration separately from the grounding and tracking modules. The three convolutional layers in the score prediction module have $D = 512, 256, 6$ output channels, respectively. We train the model with RMSProp [44] and use a batch size of 32. The initial learning rate is 10^{-4} and follows a linear schedule. We fine-tune the grounding module [9] pre-trained on Flickr30K Entities [8] with training set videos. For the tracking module, we use the models released by SiamRPN++ [13] and fix the weights. The decay rate in Algorithm 1 is set to 0.998.

C. Experiment Protocols

Table I reports the tracking results on LaSOT [15] and Lingual OTB99 [1]. One-stage grounding [9] is used for "grounding" and SiamRPN++ [13] is used for "tracking" in all reported results expect the original LSAN [1]. We list in the "Integration guidance" column the different integration methods. The **top portion** of Table I contains naive integration with either pre-defined scheduling rules or fixed fusion weights. Frame indexes such as "all," "first," and "fixed interval" indicate pre-defined scheduling is adopted and on which frames grounding is assigned higher importance. The **bottom portion** of the table contains the results of our

TABLE I
TRACKING BY LANGUAGE RESULTS ON LASOT [15] AND LINGUAL OTB99 [1]

Method	Integration Guidance (**see Sec. VI-C for detail**)	LaSOT		Lingual OTB99	
		Success	Precision	Success	Precision
Single Module and Simple Combination Baselines					
Visual grounding	All	0.416	0.411	0.442	0.551
First frame tracking	First	0.331	0.301	0.421	0.551
Middle frame tracking	Middle	0.369	0.345	0.432	0.516
Last frame tracking	Last	0.307	0.277	0.448	0.540
Random frame tracking	Random	0.361	0.328	0.434	0.514
Fixed interval tracking	Fixed interval=5	0.423	0.420	0.449	0.552
Fixed interval tracking	Fixed interval=10	0.422	0.418	0.449	0.554
Fixed interval tracking	Fixed interval=20	0.420	0.412	0.449	0.556
LSAN [1]	Fixed weights fusion	-	-	0.259	-
LSAN++ [1]	Fixed weights fusion	0.404	0.405	0.449	0.548
Feng *et al.* [22]	Tracking by detection	0.28	0.28	0.54	**0.78**
Different Variations of Our Methods					
Ours-Grounding score	Max grounding score	0.450	0.450	0.532	0.657
Ours-R score	Max R-score	0.474	0.467	0.565	0.706
Ours-RT scores	Max RT-scores	**0.478**	**0.476**	**0.581**	0.732

adaptive integration methods. The types of adopted importance scores are listed in "Integration guidance."

Various baselines and state-of-the-art methods are experimented and compared. To be specific, we systematically study the following settings:

- **Visual grounding.** One could attempt to approach tracking by language by processing each frame independently by grounding. One-stage visual grounding [9] is adopted for the experiment.
- **First frame tracking.** By taking the grounded region in the first frame as the tracking template, tracking by language is converted to a object tracking problem. This baseline is referred to as "First frame tracking."
- **Middle/ Last/ Random frame tracking.** We initialize the tracker with the grounded region in the middle, last or one random sampled frame.
- **Fixed interval tracking.** In this baseline, "grounding" is assigned a higher importance with a fixed temporal interval. We design the fixed interval to be similar to the averaged frequency of our adaptive integration.
- **LSAN/ LSAN++.** We compare to the state-of-the-art tracking by language method LSAN [1]. For a fair comparison, we strength LSAN with stronger grounding [9] and tracking [13] backbones used in other experiments, and refer to it as "LSAN++."
- **Ours-Grounding/ R/ RT scores.** We experiment with different variations of our methods. "Ours-" indicates that the GTI implementation in Section V is adopted, with different importance score selections.

D. Tracking by Language Results

1) Lingual OTB99: As the single module baseline, we first benchmark the "grounding" and "tracking" modules adopted in all following experiments. "Grounding" alone generates a success score of `0.442`, and "tracking" obtains a comparable success score of around `0.434`, as shown in "Visual grounding" and "First/ middle/ last/ random frame tracking."

"Integration" aims to improve the tracking by language performance by synergistically combining the two modules. The top portion of Table I shows several simple combinations. Fixed temporal scheduling is one possible solution that switches between grounding and tracking with a fixed interval. "Fixed interval tracking" obtains a success score of `0.449` and slightly outperforms the single module baseline. "LSAN" [1] fuses the two modules' predictions with a fixed weight applied in all frames. With the strengthened backbones, "LSAN++" generates a success score of `0.449`. In short, we observe limited improvements of less than `0.01` over the single module baseline on all simple integration methods. The limited improvements confirm that the "integration" task is nontrivial, and that the *simple combination methods are ineffective*.

Our *first* contribution is proposing the new GTI framework, where we address "integration" as a score-guided self-judging process. The comparison between the top and bottom portions of Table I shows the importance of guiding integration with the scores predicted from the corresponding frame, language query, and box. One natural choice of the score is the grounding confidence. "Ours-Grounding score" reports a success score of `0.532`, which is significantly better than the grounding baseline (`0.442`) and the simple integration (`0.449`). The improvements show the advantage of score-guided integration instead of the simple combinations.

Our *second* contribution is proposing better integration scores. As shown in the bottom portion of Table I, the "Ours-R score" achieves a success score of `0.565`, compared to `0.532` by "Ours-Grounding score," `0.449` by "LSAN++," and `0.449` by "fixed interval tracking." By jointly considering the template quality score, we further improve the success score. T-score alone does not work well because of the loss of region correctness information.

2) LaSOT: "Grounding" provides a baseline success score of `0.416`. The tracking baseline has a lower performance of `0.361`. "Tracking" performs relatively worse on LaSOT than OTB99 because the longer averaged video length in LaSOT makes tracking more challenging. For the same reason, updating the template multiple times performs better than a single template frame (cf. different intervals in "Fixed interval tracking"). To eliminate the influence of the template update frequency, we design "Fixed interval tracking" to have a similar frequency as our RT-integration, which ranges from 5 to 20 frames. "Ours-Grounding/ R/ RT score" updates

the template every 17.0/20.6/23.5 frames on LaSOT and 7.9/13.8/16.6 frames on Lingual OTB99. By eliminating the influence of the template update frequency, we show our adaptive integration performs better purely by more effective combining grounding and tracking.

We draw from LaSOT largely the same observation on "integration" as from Lingual OTB99. The simple integration methods such as "LSAN++" and "fixed interval tracking" show limited improvements over the single module baseline, while our adaptive integration significantly improves the performance. Our RT-integration achieves a success score of `0.478`, compared to `0.404` by LSAN++ and `0.423` by fixed interval tracking. Clearly, the improvement shows the importance of score-guided integration and the effectiveness of our RT-integration.

The experiments in Table I are conducted on the disambiguated LaSOT dataset, except the original LSAN [1] and Feng *et al*. [22]. To examine the benefit of LaSOT query cleaning, we compare our full model "Ours-RT score" to the one trained on original LaSOT queries. "Ours-RT score" trained and tested with the original LaSOT queries generates a success score of `0.475` and precision score of `0.469`. Compared to the performance of `0.478` and `0.476` after cleaning, the improvement is marginal. We expect the provided disambiguated queries open up the possibility of further improving tracking by language in future studies.

3) Lingual ImageNet Videos: We find that the Lingual ImageNet videos dataset is a special easy case, where current visual grounding methods already performs better than tracking by boxes ("Visual grounding" success score: `0.864`, "SiamRPN++ [13]": `0.768`]). In Lingual ImageNet videos, the target objects are mostly in the center of the frame with few distracting objects exist, which makes the task easy for visual grounding. Despite the good results on this specific dataset, such videos are far from real and oversimplify the tracking by language problem.

4) Inference Speed: A fast inference speed is important for tracking by language. We evaluate the inference speed of our GTI implementation on a desktop with Intel Core i9-9900K@3.60GHz and NVIDIA 1080TI. Our framework runs at around **20 fps**, where the grounding module takes 20ms and the tracking module takes 30ms. The proposed RT-integration module takes less than **1ms** by reusing the visual-textual features from grounding.

E. Qualitative Results Analyses

In this section, we compare the success and failure cases of the methods with naive integration modules as well as ours, to show the significance of our proposed RT-integration. We show representative examples in Figure 6. First, our method (silver boxes) are more stable and accurate when compared to per-frame visual grounding outputs (blue boxes). Including "tracking" (dark grey boxes) generates more stable results by exploiting the cross-frame visual similarity. However, the grounded region for tracker initialization in a randomly selected frame might be incorrect and thus fails "tracking" in the following frames. Figures 6 (a) and (b) show failure cases for the "First frame tracking" that our method can solve. Figures 6 (c) and (d) present challenging cases where "grounding" fails in most frames. When all compared methods fail, our RT-integration successfully combines grounding and tracking to provide mostly correct tracking results throughout the video. Overall, our proposed approach performs better by more effectively integrating grounding with tracking.

TABLE II
ORACLE ANALYSES OF TRACKING BY LANGUAGE OR GT BOXES

Method	Object referring	LaSOT Succ.	LaSOT Prec.	Lingual OTB99 Succ.	Lingual OTB99 Prec.
MEEM [45]	GT box	0.257	0.227	0.491	0.725
HCFT [46]	GT box	0.250	0.241	0.518	0.778
ECO [12]	GT box	0.324	0.301	0.675	0.888
SiamFC [19]	GT box	0.336	0.339	-	-
VITAL [47]	GT box	0.390	0.360	-	-
MDNet [48]	GT box	0.397	0.373	-	-
SiamRPN [14]	GT box	0.449	0.435	0.687	0.897
SiamRPN++ [13]	GT box	0.496	0.489	**0.698**	**0.899**
Ours-R-oracle	Language	0.624	0.656	0.645	0.826
Ours-RT-oracle	Language	**0.631**	**0.665**	0.672	0.863

Despite the effectiveness of our proposed integration, when grounding fails on all frames, there is no hope to get correct results (cf. Figure 6 (e)). RT-score estimation may also be incorrect. Figure 6 (f) shows an example that could be corrected, while our method fails to predict the correct RT-scores and correct the errors. Such failures are the cause of the gap from the oracles in Table II.

Furthermore, we experiment with referring to the same object with different language queries. We generate additional testing queries that describe different aspects of the target, *e.g*., color, location, or the relationship with other objects. Figure 7 shows good qualitative results that the method generalizes well onto free-form referring queries.

F. Oracle Analyses

Tracking by language is generally more challenging than the conventional tracking by gt box setting and tends to perform worse on the same video [1]. The proposed RT-integration greatly improves the tracking by language performance. However, the score prediction module meanwhile introduces new errors and potentially limits the overall performance. In this section, we examine the upper bound of the GTI framework that has an ideal "integration" model, given the status quo of grounding [9] and tracking [13]. We compare the oracles to both tracking by language and by gt box results, specifically the following settings:

- **Tracking by gt box.** With the same dataset split, tracking by ground-truth box [12]–[14], [19], [47], [48] serves as an upper bound of tracking with ideal target specifications.
- **Ours-R-oracle.** We design two oracle analyses with the same GTI implementation in Section V. The R-score in the oracle analyses is calculated with the ground-truth box at each frame instead of predicted.
- **Ours-RT-oracle.** "Ours-RT-oracle" considers both the region correctness and template quality scores.

As shown in Table II, the GTI framework with an ideal integration module achieves comparable (on shorter videos [1]) if

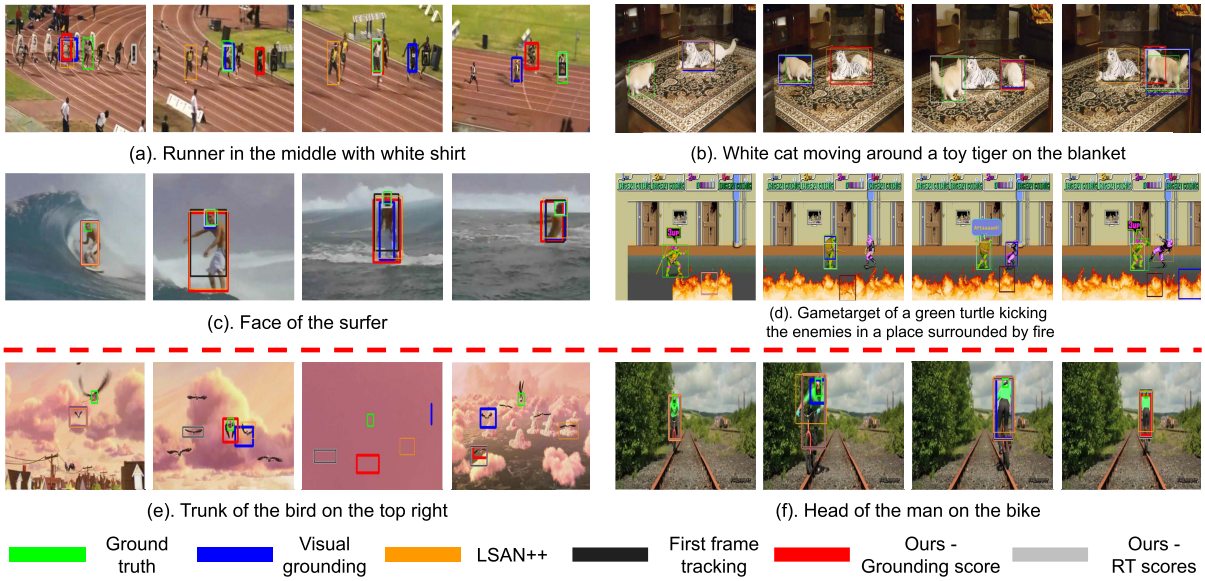

Fig. 6. Representative success cases (top two rows) and failures (bottom row) of our method. Figure (b), (d) are from LaSOT and the others are from Lingual OTB99. Best viewed in color and zoomed in.

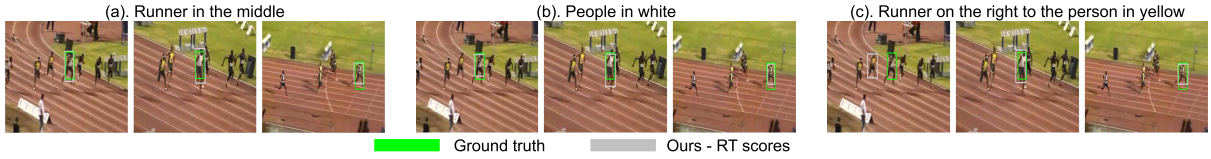

Fig. 7. Qualitative examples of referring to the same object with different language queries at inference.

not better (on longer videos [15]) performance than the state-of-the-art tracker [13]. The good oracle performance implies the possibility of tracking by language to achieve comparable results to tracking by gt box, despite the more challenge setting. Meanwhile, the existing gap between the performance of the oracle GTI and our implementation shows that the integration problem is non-trivial, and motivate us to develop better integration methods in future studies. Finally, with the continuously improving grounding and tracking methods, we expect the future GTI frameworks with stronger modules to further improve the tracking by language performances.

G. Ablation Study

In this section, we conduct ablation studies to understand our method better. We first compare alternative adaptive integration methods to the hard switch approach introduced in Section IV-C. We then show that our proposed "integration" module's importance and effectiveness hold under different "grounding" and "tracking" backbones.

1) Adaptive Integration: Given the obtained scores for integration, there are alternative methods to the hard switch approach described in Section IV-C. We experiment with other adaptive integration methods to examine their influences on the performance. We conduct the ablation studies on adaptive integration with the oracle RT-scores detailed in Section VI-F to eliminate the influence of score prediction quality.

Given the predicted tracking results, grounding results, and integration scores, the final step is to integrate the tracking and grounding prediction with the predicted scores for both tracking template update and current frame prediction. We first explore three alternative ways of updating the tracking template. Our adopted "`greedy update`" option outputs the grounding prediction and updates the tracking template whenever a higher score appears. "`Improvement threshold`" follows the same greedy update protocol with a tuned score improvement threshold of 20% included. "`Fixed weight update`" consists of a memory module and updates the template's visual feature with a fixed update rate of 0.9 [49]. "`Score weighted update`" further adopts the predicted scores as the update rate.

Table III shows the success and precision scores of the compared adaptive integration methods. Our adopted approach generates a success score of `0.672`, which is comparable to the best score of `0.675`. We observe no significant gain by more complex alternatives and thus choose the simple yet effective "`greedy update`" as the adopted approach.

Other than the tracking template update, the adaptive integration method also generates the final prediction at each frame. Our adopted "hard switch" method that outputs either tracking or grounding results based on the integration scores. As an alternative, we experiment with the "`soft fusion`" used by previous studies [1], where the output fusion is computed as a weighted sum of the grounding and tracking heatmaps with the predicted per-frame integration score. We observe the "hard switch" outperforms the "soft fusion" and thus adopt the "hard switch" approach for output fusion.

2) GTI Backbones: We then experiment with the influence of "integration" with different "grounding" and "tracking" backbones. We replace the backbones with relatively weaker (but faster) modules and benchmark the corresponding GTI

TABLE III

TRACKING BY LANGUAGE RESULTS WITH DIFFERENT ADAPTIVE INTEGRATION METHODS ON LINGUAL OTB99. THE "GREEDY UPDATE" AND "HARD SWITCH" RESULT SHOWN IN THE FIRST ROW IS THE APPROACH WE ADOPTED. WE HIGHLIGHT THE BEST THE SECOND-HIGHEST SCORES BY **BOLD** AND UNDERLINE, RESPECTIVELY

Template update	Output fusion	Succ.	Prec.
Greedy update	Hard switch	0.672	0.863
Greedy update	Soft fusion	0.646	0.843
Improvement threshold	Hard switch	0.632	0.814
Fixed weight update	Hard switch	**0.675**	**0.867**
Score weighted update	Hard switch	0.668	0.856

TABLE IV

TRACKING BY LANGUAGE RESULTS WITH DIFFERENT GROUNDING AND TRACKING BACKBONES ON LINGUAL OTB99. COMPARING DIFFERENT INTEGRATION METHODS WITH THE SAME GROUNDING AND TRACKING MODULE, THE EFFECTIVENESS OF THE PROPOSED "OURS-RT SCORES" HOLDS UNDER DIFFERENT BACKBONES

Integration method	Grounding	Tracking	Lingual OTB99	
			Succ.	Prec.
Visual Grounding	Onestage-light	None	0.379	0.491
Visual Grounding	Onestage	None	0.442	0.551
Fixed interval tracking	Onestage-light	SiamRPN++	0.391	0.492
Fixed interval tracking	Onestage	SiamRPN	0.446	0.553
Fixed interval tracking	Onestage	SiamRPN++	0.449	0.554
Ours-RT scores	Onestage-light	SiamRPN++	0.570	0.723
Ours-RT scores	Onestage	SiamRPN	0.555	0.701
Ours-RT scores	Onestage	SiamRPN++	0.581	0.732

implementations. We replace the adopted SiamRPN++ [13] with SiamRPN [14], and one-stage visual grounding [9] with a lighter version Onestage-light [9].

Table IV shows the obtained results. In short, better grounding and tracking modules generally lead to better tracking by language performances. More importantly, our proposed "RT-Integration" brings significant success score improvements with different backbones (*cf*. "Fixed interval tracking " and "Ours-RT scores" with the same backbone). The consistent improvements of 0.179, 0.109, 0.132 over the simple combination baseline indicate that "RT-integration" is effective under different grounding and tracking backbones. With the continuously improving grounding and tracking methods, we expect future GTI implementations to further improve the tracking by language performance.

VII. CONCLUSION

We have proposed a new GTI framework for tracking by language where we decompose the task into three sub-tasks: grounding, tracking, and integration. We focus on the key sub-task of "integration" that synergistically combines grounding and tracking, and propose an RT-integration module that defines two scores to guide integration in each frame. The R-score represents the region correctness, and the T-score represents the template quality. We benchmark our real-time implementation of the GTI framework on LaSOT and Lingual OTB99 to demonstrate highly promising results.

REFERENCES

[1] Z. Li, R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders, "Tracking by natural language specification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6495–6503.

[2] M. Kristan *et al.*, "A novel performance evaluation methodology for single-target trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016.

[3] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[4] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[5] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[6] M. Yamaguchi, K. Saito, Y. Ushiku, and T. Harada, "Spatio-temporal person retrieval via natural language queries," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1453–1462.

[7] J. Lei, L. Yu, M. Bansal, and T. Berg, "TVQA: Localized, compositional video question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1369–1379.

[8] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting Region-to-Phrase correspondences for richer Image-to-Sentence models," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 74–93, May 2017.

[9] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, and J. Luo, "A fast and accurate one-stage approach to visual grounding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4683–4693.

[10] L. Yu *et al.*, "MAttNet: Modular attention network for referring expression comprehension," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1307–1315.

[11] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 69–85.

[12] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

[13] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[14] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[15] H. Fan *et al.*, "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5374–5383.

[16] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.

[17] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. D. Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1090–1097.

[18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[19] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 850–865.

[20] J. Shi, J. Xu, B. Gong, and C. Xu, "Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10444–10452.

[21] X. Wang, C. Li, R. Yang, T. Zhang, J. Tang, and B. Luo, "Describe and attend to track: Learning natural language guided structural representation and visual attention for object tracking," 2018, *arXiv:1811.10014*. [Online]. Available: http://arxiv.org/abs/1811.10014

[22] Q. Feng, V. Ablavsky, Q. Bai, G. Li, and S. Sclaroff, "Real-time visual object tracking with natural language description," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 700–709.

[23] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "ReferItGame: Referring to objects in photographs of natural scenes," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 787–798.

[24] B. A. Plummer, P. Kordas, M. Hadi Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik, "Conditional image-text embedding networks," in *Proc. ECCV*, 2018, pp. 249–264.

[25] A. Sadhu, K. Chen, and R. Nevatia, "Zero-shot grounding of objects from natural language queries," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4694–4703.

[26] L. Wang, Y. Li, J. Huang, and S. Lazebnik, "Learning two-branch neural networks for image-text matching tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 394–407, Feb. 2019.

[27] L. Yu, H. Tan, M. Bansal, and T. L. Berg, "A joint Speaker-Listener-Reinforcer model for referring expressions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7282–7290.

[28] X. Chen, L. Ma, J. Chen, Z. Jie, W. Liu, and J. Luo, "Real-time referring expression comprehension by single-stage grounding network," 2018, *arXiv:1812.03426*. [Online]. Available: http://arxiv.org/abs/1812.03426

[29] Z. Yang, T. Chen, L. Wang, and J. Luo, "Improving one-stage visual grounding by recursive sub-query construction," in *Proc. ECCV*, 2020, pp. 1–17.

[30] Z. Zhang, Z. Zhao, Y. Zhao, Q. Wang, H. Liu, and L. Gao, "Where does it exist: Spatio-temporal video grounding for multi-form sentences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10668–10677.

[31] A. Sadhu, K. Chen, and R. Nevatia, "Video object grounding using semantic roles in language description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10417–10427.

[32] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

[33] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 784–799.

[34] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner, "Precise detection in densely packed scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5227–5236.

[35] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6409–6418.

[36] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[37] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 817–825.

[38] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang, "Object detection in videos by high quality object linking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1272–1278, May 2020.

[39] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[40] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[41] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6182–6191.

[42] Y. Lu, T. Wu, and S. Chun Zhu, "Online object tracking, learning and parsing with and-or graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3462–3469.

[43] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[44] T. Tieleman and G. Hinton, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[45] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 188–203.

[46] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082.

[47] Y. Song *et al.*, "VITAL: VIsual tracking via adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8990–8999.

[48] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.

[49] T. Yang and A. B. Chan, "Learning dynamic memory networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 152–167.

Zhengyuan Yang (Graduate Student Member, IEEE) received the B.E. degree in electrical engineering from the University of Science and Technology of China in 2016. He is currently pursuing the Ph.D. degree in computer science with the University of Rochester, Rochester, NY, USA, advised by Prof. Jiebo Luo. His research interests include vision+language and multimodal learning.

Tushar Kumar received the B.E. degree in information technology from the Delhi College of Engineering in 2013 and the master's degree in computer sciences from the University of Rochester in 2020.

Tianlang Chen (Graduate Student Member, IEEE) received the B.E. degree in electrical engineering from the University of Science and Technology of China in 2016. He is currently pursuing the Ph.D. degree with the Computer Science Department, University of Rochester, under the supervision of Prof. Jiebo Luo. His research interests mainly include tasks related to joint visual-textual learning, such image captioning, image-text matching, visual question answering, and visual grounding.

Jingsong Su received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in July 2011. He is currently an Associate Professor with the Software School, Xiamen University. His research interests include natural language processing and neural machine translation.

Jiebo Luo (Fellow, IEEE) is currently a Professor of Computer Science with the University of Rochester, where he joined in 2011 after a prolific career of 15 years at the Kodak Research Laboratories. He has authored over 400 technical articles and holds over 90 U.S. patents. His research interests include computer vision, NLP, machine learning, data mining, computational social science, and digital health. He is a Fellow of ACM, AAAI, SPIE, and IAPR. He has been involved in numerous technical conferences, including serving as the Program Co-Chair of ACM Multimedia 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, as well as the General Co-Chair of ACM Multimedia 2018. He has served on the editorial boards for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON MULTIMEDIA (TMM), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), IEEE TRANSACTIONS ON BIG DATA (TBD), *ACM Transactions on Intelligent Systems and Technology* (TIST), *Pattern Recognition*, *Knowledge and Information Systems* (KAIS), *Machine Vision and Applications*, and *Journal of Electronic Imaging*. He is current the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA.