# Real-time Visual Object Tracking with Natural Language Description

Qi Feng[1], Vitaly Ablavsky[1], Qinxun Bai*[2], Guorong Li[3], and Stan Sclaroff[1]

[1]Boston University, {fung,ablavsky,sclaroff}@bu.edu
[2]Horizon Robotics, qinxun.bai@gmail.com
[3]University of Chinese Academy of Sciences, liguorong@ucas.edu.cn

## Abstract

*In this work, we argue that conditioning on the natural language (NL) description of a target provides information for longer-term invariance, and thus helps cope with typical tracking challenges. However, deriving a formulation to combine the strengths of appearance-based tracking with the language modality is not straightforward. Therefore, we propose a novel deep tracking-by-detection formulation that can take advantage of NL descriptions. Regions that are related to the given NL description are generated by a proposal network during the detection stage of the tracker. Our LSTM based tracker then predicts the update of the target from regions proposed by the NL based detection stage. Our method runs at over 30 fps on a single GPU. In benchmarks, our method is competitive with state of the art trackers that employ bounding boxes for initialization, while it outperforms all other trackers on targets given unambiguous and precise language annotations. When conditioned on NL descriptions only, our model doubles the performance of the previous best attempt [25].*

## 1. Introduction

Progress in visual object tracking has been driven in part by the establishment of standard datasets and competitions/challenges [20, 34] tied to common evaluation metrics [10, 34]. These datasets range from specific scenarios, such as video surveillance [11], containing predominantly pedestrians and vehicles captured with stationary cameras, to datasets that depicts objects of unrestricted categories captured with cameras undergoing arbitrary motion. Most recent approaches focused on the latter types of datasets, proposing new types of deep features [2, 22, 23, 36] that are robust to occlusions, rapid motion, etc.

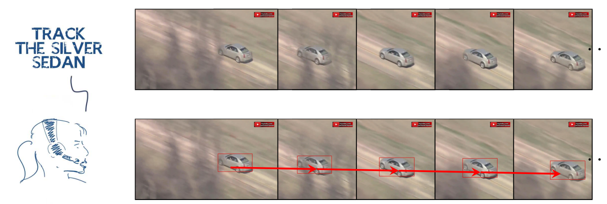



Figure 1: The goal is to perform tracking by natural language specifications given by a human. For example, someone specifies *track the silver sedan running on the highway* and our goal is to predict a sequence of bounding boxes on the input video. We also take advantage of the natural language to better handle the cases of occlusion and rapid motion of the target throughout the tracking process.

However, since the "unrestricted categories" datasets require good tracking performance across a wide range of scenarios, and this affects how the tracking problem itself is posed. Specifically, it has been a *de facto* standard to initialize a tracker with a bounding box in the first frame, and require that this bounding box accurately cover the object of interest. Such a requirement stands in contrast to tracking in the surveillance domain, where a tracker is expected to self-initialize using proposals for regions likely to contain objects of interest [11].

While it may be acceptable, for the sake of benchmark evaluation, to condition a tracker on an initial bounding box, it limits the practical applicability of the tracker. In particular, it assumes forensic-style scenarios where a human operator sitting at a workstation with a high-definition monitor has the time and attention to specify an exact bounding box for each object of interest. Yet, as demand for computer vision evolves, so do requirements and assumptions for how vision algorithms fit within the intended applications. For ex-

*Work done at Hikvision Research America ample, a drone operator with a tablet would be unable to draw a precise bounding box on an object the drone should circle around, *e.g.*, a vehicle speeding down a highway. In such a scenario, as is shown in Fig. 1, it is more acceptable for the operator to speak into the head-microphone: "track the silver sedan."

Combining natural language understanding and object tracking potentially yields multiple benefits. Firstly, it has the ability to dispense with the requirement of a bounding-box initialization. Secondly, it combines the complementary nature of the two modalities, language and visual appearance, which promotes the longer-term stability of a tracker.

In spite of the apparent benefits, the role of natural language in tracking has been relatively unexplored [25]. One of the main challenges is the derivation of a formulation where a target's location is conditioned both on its natural language description and its appearance history in previous frames. Another challenge is to guarantee the computational efficiency in spite of the added complexity.

Contributions of this paper include:

- A novel tracking-by-detection formulation that is conditioned on a natural language description of the target to be tracked; in particular, we are the first to derive a tracker that employs an RNN to estimate the target's location from regions proposed by a deep network, conditioned on a language network. If desired, our formulation can be conditioned on both the NL description *and* a bounding box, thus providing added flexibility in how our tracker can be used in practice.

- Our tracker nearly doubles the performance of the best prior attempt to condition purely on natural language [25], and thus brings us closer to making NL tracking practical. It compares favorably to the state of the art trackers that employ bounding boxes for initialization in cases when NL description is precise. We show that, for detection based trackers, targets can be extended to a wider range of categories with the guidance of NL.

- We implement our tracker following the derivation in Section 3 without any additional "bells-and-whistles," such as searching across scales, reliance on non-linear dynamics to model target motion, or sub-window attention [30]. Note that the added computational complexity of [30] results in inference speed below 1 fps. Even though we process more information than traditional approaches, visual appearance *and* NL, our method is computationally efficient, yielding a tracker operating at roughly 30 fps on a single GPU.

## 2. Related Works

Regional convolution networks have seen wide success in recent years. R-CNN [13], Fast R-CNN [12], and Faster R-CNN [28] are a family of regional convolution networks developed for object detection. The introduction of the region proposal network (RPN) in the Faster R-CNN framework [28] makes the generation of object proposals efficient and fast compared to the selective search approach [31] used in R-CNN [13]. The proposal module in our tracker is built upon the RPN, which proposes regions from convolution features.

Various models have been proposed to address the problem of object detection with natural language (NL) specification [4, 16, 21, 24, 32]. Our work employs NL understanding techniques similar to some image captioning approaches to perform the task of object detection with NL specification.

In recent work, word embeddings [27] are used to represent words in numerical vector representations. In our work, word embeddings are used for measuring the similarities between NL specifications and images. Johnson et al. proposed a dense captioning model [16] based on an RPN like region proposal module for region proposals and used a bi-linear sampler to obtain a list of regional features for each of the best proposals after the RPN. The regional features are further used as the initial state in an LSTM model for captioning tasks, which is similar to the image captioning model in [32].

Tracking by detection models [3, 17] and Bayesian filtering based algorithms [5, 18] have been widely adopted. Some deep learning based models [2, 6, 7, 26, 29, 30, 33] have been introduced in recent years, and are argued to perform better when handling occlusion and appearance change. SiamFC [2] conducts a local search for regions with a similar regional visual features obtained by a CNN in every frame. ECO [7] applies convolutional filters on convolution feature maps to obtain satisfactory performance on multiple tracking datasets. ECO still suffers from efficiency issues [15], though its efficiency is improved from the original convolution filter based tracker, C-COT [8]. These trackers maintain appearance and motion models explicitly by maintaining the visual features over time. ATOM [6] uses overlap maximization to improve detection accuracy by training an additional modules both to modulate the IoU of a prediction and to predict the existance of distractors. ROLO [26] attempts to model the temporal relationships between detections across continuous frames. The RNN performs a regression task given detections as input.

In contrast, NL based tracking allows the tracker to track a wider range of objects. Li et al. introduced

| | |
|---|---|
| **Deterministic Variables** | |
| $t$ | frame index |
| $I_t$ | the $t$-th frame image from a video |
| $Q$ | the given natural language query for a video |
| $X_t^*$ | target (ground truth) bounding box. |
| **Random Variables** | |
| $F_t$ | visual feature map computed by CNN |
| $\mathbf{X}_t = \{X_{t,i}\}$ | set of bounding boxes proposed by RPN |
| $\mathbf{RF}_t = \{RF_{t,i}\}$ | set of region features for bounding boxes in $\mathbf{X}_t$ |
| $\mathbf{S}_t = \{S_{t,i}\}$ | set of similarities between region features and $Q$ |
| $\mathbf{X}_t^{\text{det}}$ | set of bounding boxes chosen by detection module |
| $\mathbf{RF}_t^{\text{det}}$ | set of region features for bounding boxes in $\mathbf{X}_t^{\text{det}}$ |
| $\hat{X}_t$ | bounding box predicted by the tracker |

Table 1: Notations of deterministic and random variables.

three models that exploit NL descriptions of the target [25], which jointly consider the regional appearance features and NL features. They proposed a way to initialize a tracker without a bounding box. However, it is 50% less accurate compared to ground truth bounding box initialized tracker.

Our model performs the same tracking by NL task in a different way by jointly performing the tracking and detection at the same time. Comparing to [25], instead of only using the NL description at the first frame to initialize the tracker, we utilize the NL description during the entire tracking process and nearly doubles its performance.

## 3. The Proposed Model

Our proposed tracker is composed of two stages. The first stage is the detection stage which contains two modules. The first module is a deep convolutional neural network that proposes candidate detection regions, and the second module is an RNN that measures the similarities between visual representations of proposed regions from the first module and the input language embeddings [27]. After the detection stage, regions are ranked by the similarities and passed into the tracking module, which is another RNN that performs tracker update from previous frame based on detection candidates. The overall architecture of our model is shown in Fig. 2, expressed as a Bayesian network, while the detection stage is decoupled from the tracker and is detailed in Fig. 3.

In Sec. 3.1, we formulate the detection problem and compute the posterior probability of a detection given the target's NL description. In Sec. 3.2, we present our tracking module. Finally, we present implementation details in Sec. 3.3. Notations are summarized in Table 1.

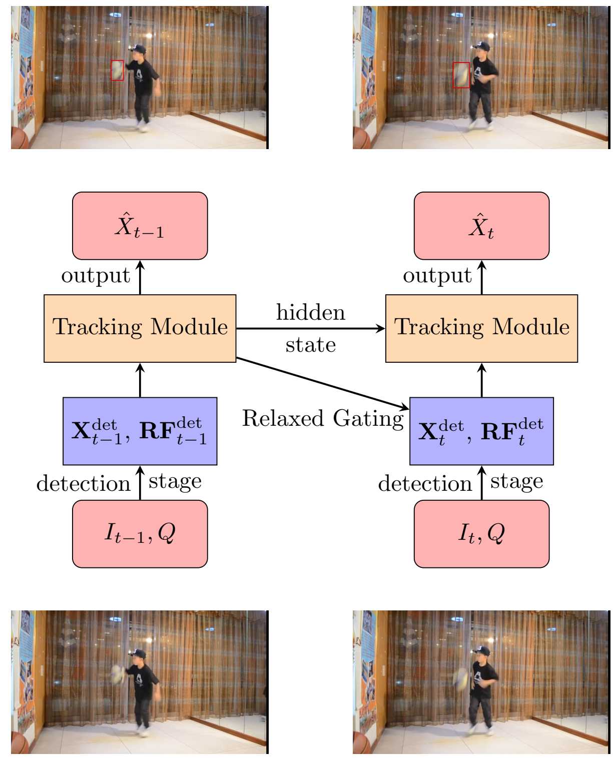



Figure 2: The proposed model. Red rounded rectangles are inputs and outputs of the model. The detection stage (detailed in Fig. 3) of the model takes frames and NL descriptions as inputs and returns a set of candidate bounding boxes and their corresponding region features. The tracking module performs tracking by detection. Predictions are then utilized to assist the detection stage with relaxed gating.

### 3.1. Detection

Let $I_1, \ldots, I_T$ denote the sequence of video frames, where $T$ is the total number of frames. Let $Q$ be the NL description of the target. Let $\mathbf{X}_t = \{X_{t,i}\}_{i=1}^n$ be the set of candidate bounding boxes.

Given frame $I_t$, a deep convolutional neural network is utilized to extract the visual feature map, which we denote as $F_t \in \mathbb{R}^{H \times W \times D}$, where $H, W,$ and $D$ are height, width, and depth of the visual feature map.

A region of interest (RoI) pooling on $F_t$ is then adopted to obtain regional convolution features of a uniform size $A \times A \times D$ for all bounding boxes in $\mathbf{X}_t$. We denote the set of regional convolution features as $\mathbf{RF}_t = \{RF_{t,i}\}_{i=1}^n$, where $RF_{t,i} \in \mathbb{R}^{A \times A \times D}$.

We adopt a Region Proposal Network (RPN) that performs class agnostic object detection on $\mathbf{RF}_t$. We consider this RPN as an estimation of a conditional probability $\Pr[X_{t,i}|I_t]$.

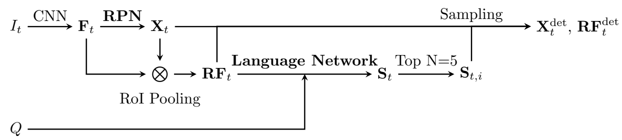



Figure 3: The architecture of the detection stage, which consumes inputs (frame $I_t$ and natural language description $Q$) and produces bounding boxes $\mathbf{X}_t^{\text{det}}$ and their corresponding regional convolution features $\mathbf{RF}_t^{\text{det}}$ chosen by the RPN and the language network. The detection stage is the first stage in our model, preceding the tracking stage in Fig. 2.

Similarly to the RPN, we design a language network that estimates the similarity between $Q$ and each $X_{t,i}$, which can be regarded as the conditional probability $\Pr[X_{t,i}|Q]$ for each bounding box.

Our language network is an LSTM network that takes word embeddings from GloVe [27] as inputs. The language network learns a sentence embedding of size 300, which is also the size of the GloVe embedding. The language module then produces a similarity score $S_{t,i}$ between the sentence embedding from $Q$ and $RF_{t,i}$.

We train our language network as a regression task with a log loss. The target similarity $\hat{S}_{t,i}$ is the intersection-over-union (IoU) between $X_{t,i}$ and the ground truth bounding box $X_t^*$. That is

$$L_{\text{lang}}(t) = -\sum \left(\hat{\mathbf{S}}_t \log(\mathbf{S}_t) + (1-\hat{\mathbf{S}}_t)\log(1-\mathbf{S}_t)\right), \tag{1}$$

where

$$\hat{\mathbf{S}}_t = \text{IoU}(\mathbf{X}_t, X_t^*).$$

After estimating the joint conditional probability $\Pr[X_{t,i}|Q, I_t]$ for all candidates $X_{t,i} \in \mathbf{X}_t$ proposed by the RPN following the Bayes rule, we rank the bounding boxes by these new scores. We choose a set of the top $N = 5$ bounding boxes and denote this set as $\mathbf{X}_t^{\text{det}}$. The detection set for each frame $\mathbf{X}_1^{\text{det}}, \ldots, \mathbf{X}_T^{\text{det}}$, and their corresponding sets of regional convolution features $\mathbf{RF}_1^{\text{det}}, \ldots, \mathbf{RF}_T^{\text{det}}$ are further pipelined into the tracking module to perform tracking sequentially.

The architecture of the detection stage is summarized in Fig. 3.

### 3.2. Tracking

A recurrent neural network (RNN) is well-suited for object tracking due to its ability to maintain model states. In particular, we use a long short-term memory (LSTM) architecture to maintain the motion, position and appearance history of the target in its hidden-state representation. Detections $\mathbf{X}_t^{\text{det}}$ and their corresponding regional convolution features $\mathbf{RF}_t^{\text{det}}$ are recurrently fed into the tracking module as inputs. We apply a fully connected layer with four neurons on LSTM outputs to obtain the track for time $1, \ldots, T$. We use $\hat{X}_t$ to denote the tracker prediction at frame $I_t$. The tracking module is trained as a regression task with smoothed L1 loss [28] for each time step.

**Relaxed Gating:** During inference of the detection stage of our tracker, we apply a relaxed gating to $\mathbf{S}_t$ within the detection module.

We use the difference between $\hat{X}_{t-1}$ and $\hat{X}_{t-2}$ to measure the velocity $v$ for our target. We measure the distance $\mathbf{d}_t$ from $\hat{X}_{t-1}$ and boxes from detection module $\mathbf{X}_t^{\text{det}}$. Similarities from the language module is then updated to "reflect" gating on the motion.

$$\mathbf{S}_t^{\text{relaxed gating}} \leftarrow \mathbf{S}_t \cdot v/\mathbf{d}_t. \tag{2}$$

Relaxed gating is applied to bounding box proposals during evaluation and training of the tracking module. The goal of relaxed gating is to suppress signals from objects that are related to the NL description but are not in the target's trajectory.

### 3.3. Implementation Details

We train and test our detection and tracking modules with original resolution images and NL annotations. We crop images so that the longest side of the image is 1333 pixels [14]. For anchors, we use 5 different sizes, $[32^2, 64^2, 128^2, 256^2, 512^2]$, which is slightly larger than the original Faster-RCNN [28], to accommodate the VisualGenome [19] dataset. Three different aspect ratios of 1:1, 1:2, and 2:1 are used.

We adopt a three stage training algorithm to learn the parameters in our RPN, language network, and tracking module. The training algorithm is shown in Alg. 1. In the first stage, we train the RPN as described in Faster RCNN [28] using the Visual Genome Dataset [19]. We use the same loss used in Faster RCNN. After the RPN is trained, we freeze parameters in the RPN and train the second stage, our language network, using the loss described in Sec. 3.1. At this point, our model is trained to propose candidate regions that are related to the NL description, *i.e.* $\mathbf{X}_t^{\text{det}}$. The last stage entails training the tracking module. As we freeze parameters in both the RPN and the language network, we apply relaxed gating to the RPN and language detection as described in Sec. 3.2. Region convolution features together with bounding box predictions are fed into our tracking module. The tracking module, described in Sec. 3.2, is finally trained as a regression task on a smoothed L1 loss [28].

The algorithm for inference is shown in Alg. 2. If the initial bounding box is not given, the RPN, the language network and the tracking module work the

**Algorithm 1:** Curriculum training for the tracker.

```
while Training RPN do
    X_t ← F_RPN(I_t);
    L_RPN ← RPN Loss(X_t, X*_t);
    Backprop(L_RPN);
FixParameters(F_RPN);
while Training Language Network do
    X_t ← F_RPN(I_t);
    X^det_t, S_t ← F_lang(X_t);
    L_lang ← LangNet Loss(X^det_t, X*_t, Q, S_t);
    Backprop(L_lang);
FixParameters(Language Network);
while Training Tracker do
    for t ∈ {1, ···, T} do
        X^det_t, RF^det_t ← Top-K(X_t);
        X̂_t ← F_tracker(X^det_t, RF^det_t);
    L_tracker ←
      Σ_{t∈{1,2,···,T}} Tracker Loss(X̂_t, X*_t);
    Backprop(L_tracker);
```

**Algorithm 2:** Inference using the proposed tracker.

```
if Initial Bounding Box is Known then
    X̂_1 ← X*_1;
else
    X_1 ← F_RPN(I_1);
    RF_1 ← ROI Pooling(X_1, F_1);
    S_1 ← F_lang(RF_1, Q);
    ReRank(X_1, S_1);
    X^det_1, RF^det_1 ← Top-K(X_1);
    X̂_1 ← F_tracker(X^det_1, RF^det_1);
for I_t ∈ {I_2, I_3, ···, I_T} do
    X_t ← F_RPN(I_t);
    RF_t ← ROI Pooling(X_t, F_t);
    S_t ← F_lang(RF_t, Q);
    S_t ← Relaxed Gating(X̂_{t−1}) · S_t;
    ReRank(X_t, S_t);
    X^det_t, RF^det_t ← Top-K(X_t);
    X̂_t ← F_tracker(X^det_t, RF^det_t);
Result: X̂_1, ···, X̂_T
```

same way as in other frames, except for the fact that no relaxed gating is applied to initialize the tracker.

We use a learning rate starting at 0.001 and AdaGrad [9] for training the model. The proposed tracker is trained with a TensorFlow [1] 1.13 implementation.

# 4. Experiments

The goal of our experiments is to demonstrate the advantage of our proposed approach in utilizing NL descriptions either independently or jointly with appearance cues in visual tracking. We first compare our model against the best published NL-based tracker [25] in Section 4.2. In Section 4.3, we demonstrate that with the help of language cues a tracker can reacquire targets and boost performance, with comparisons between state of the art trackers that employ bounding boxes for initialization. Finally, in Section 4.4, we conduct ablation studies on our tracker.

## 4.1. Datasets and Evaluation Criteria

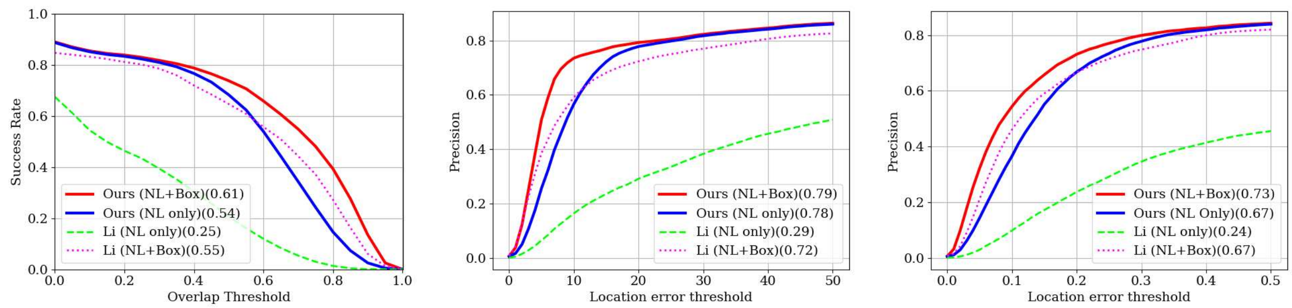
Our problem statement of combining NL description and appearance for tracking [25] differs from the majority of prior published works [2, 8, 30]. This opens the possibility for new ways of evaluating trackers but also presents challenges in comparing to previous trackers.

In particular, these challenges stem from the fact that a video clip can be labeled with a diverse set of NL descriptions. These descriptions can range from precise, *e.g.*, “a sliver sedan” to vague, *e.g.*, “a monkey among other monkeys.” Handling ambiguous or conflicting descriptions is outside the scope of this work.

We are aware of two publicly-available tracking datasets cited in [10] and [25] that have NL annotations defining the target to track. These datasets exhibit biases in terms of the linguistic style of the annotations, and how precise they are in defining the target to be tracked. Nonetheless, we adopt these two datasets, LaSOT [10] and otb-99-lang [25] for our evaluation to allow an informative comparison with our baselines.

The problem of NL description bias in these datasets is prominent, both in how natural the NL description seems to a native English speaker and whether or not it uniquely defines the target specified by the ground-truth bounding box.

While the former, *i.e.*, “naturalness” of the descriptions is hard to quantify, the later, “ambiguity” of the descriptions can be tested. We set up a simple task where outsourced workers rely on the NL description of each video to select the object to be tracked for randomly sampled frames. We aggregate videos from LaSOT that pass this test into a dataset we call “NL-consistent LaSOT”. Our crowd workers determine 21 videos from LaSOT testing set are NL-consistent. We observe a decisive improvement in the effectiveness of our tracker in handling occlusions and fast motion in videos from NL-consistent LaSOT compared to LaSOT.




(a) Success Plots on OTB (b) Precision Plots on OTB (c) Norm. Precision Plots on OTB

Figure 4: Our method vs. prior best NL tracker Li [25] on OTB-99-language dataset using the OPE protocol.

| | OTB | | LaSOT | |
|---|---|---|---|---|
| **Tracker** | Suc. | Prec. | Suc. | Prec. |
| Ours (Box+NL) | 0.61 | 0.79 | 0.35 | 0.35 |
| Li [25] (Box+NL) | 0.55 | 0.72 | - | - |
| Ours (NL Only) | 0.54 | 0.78 | 0.28 | 0.28 |
| Li [25] (NL Only) | 0.25 | 0.29 | - | - |

Table 2: Comparison between best published NL based tracker Li [25], and our proposed tracker on LaSOT and OTB-99-lang with different set-ups. AUC of success plots and precision plots (shown in Fig. 4 and Fig. 5) are reported. As [25] only released code for OTB dataset, LaSOT results are omitted in this table. Box+NL stands for models that condition on both NL and ground truth bounding box at frame 1. NL-Only stands for models that condition only on NL.

We adopt standard evaluation metrics for visual object trackers: 1. One pass evaluation (OPE) of success rate at different intersection over union (IoU) area under curve (AUC), and 2. OPE of mean average precision (MAP) over location error threshold AUC.

### 4.2. Language Trackers

The contribution of this work is a tracker that exploits both visual cues and NL descriptions of the target. Li et al. [25] introduced the tracking by natural language specification problem. It is worth noting that initializing a tracker with the bounding box is not free (except when working on academic datasets) and cannot be taken for granted. Therefore, we compare the proposed model with [25] in two different setups of the problem: tracking with or without the initialization of a bounding box.

As no training codes are available for [25], we are not able to report its results on LaSOT. The results are summarized in Table 2 and Fig. 4. These results show that by adopting a tracking-by-detection framework, when trackers are conditioned on only NL, our model doubles the tracking performance of the previous best attempt [25]. Additionally, when a ground truth bounding box is jointly provided with the NL description, our model outperforms [25].

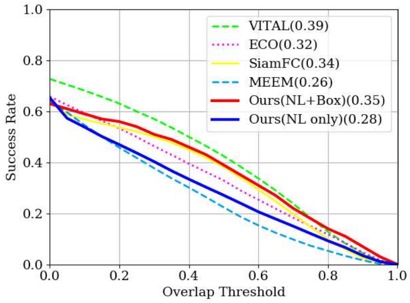
### 4.3. Comparison with Visual Object Trackers

We follow the standard evaluation procedures of OPE of success rate over IoU curve and MAP over location error threshold curve, and we compare our proposed tracker with the following trackers: Li *et al.*'s Model I [25], VITAL [30], SiamFC [2], ECO [7], and MEEM [35]. The results are summarized in Fig. 4 and Fig. 5. Our tracker is competitive with state of the art trackers.

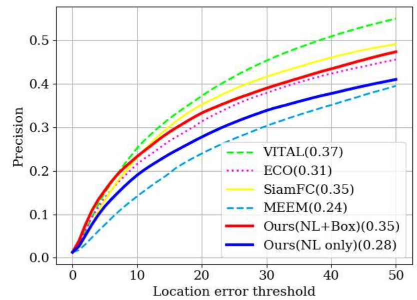
On "NL-consistent LaSOT" our tracker outperforms all other trackers by 20 percentage points on AUC of the success plots in Fig. 6. This is due to the verified NL description which makes our tracker better at detection and hence yields a better tracking performance.

**Mean IoU over Time:** In addition to these standard evaluation metrics, we also qualitatively evaluate the performance of trackers based on the OPE mean IoU over time curve. For long term tracking, the inability to re-acquire the target after losing the target during tracking is fatal to trackers. The mean IoU over time curve gives us a qualitative way of evaluating trackers in terms of long-term stability. We plot the IoU for test videos at each time step, in which the horizontal axis is the index of frame and the vertical axis is IoU.

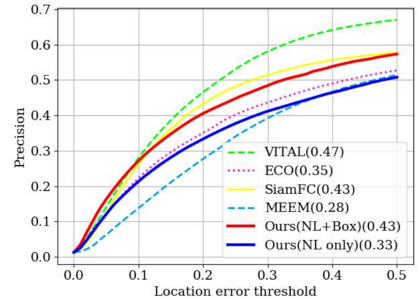
As our proposed tracker employs tracking by detection, language based detection plays an important role in the entire tracking process. An example of IoU at different time steps is shown in Fig. 7. It shows that our tracker can re-acquire the target with NL descriptions after losing the target due to cases like occlusion and/or rapid motion.



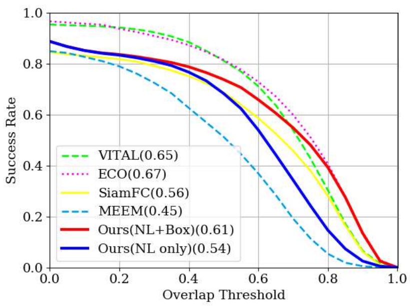

(a) Success Plots on LaSOT



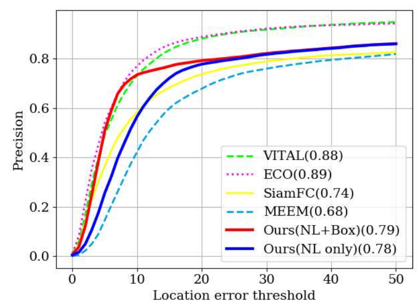

(b) Precision Plots on LaSOT



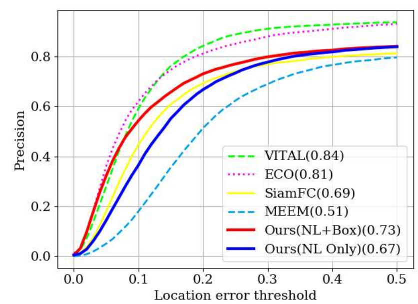

(c) Norm. Precision Plots on LaSOT




(d) Success Plots on OTB

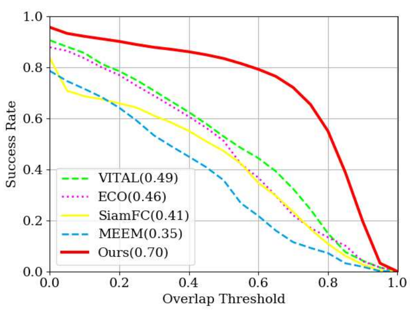



(e) Precision Plots on OTB

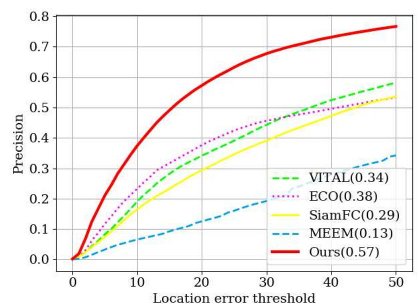



(f) Norm. Precision Plots on OTB

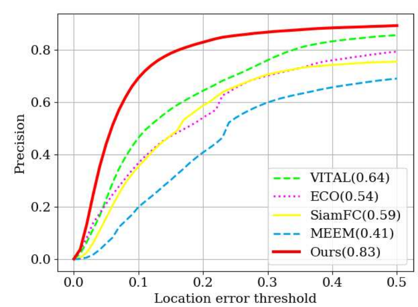
Figure 5: Comparison of our approach with state-of-the-art traditional trackers: one pass evaluation success rate at different intersection over union thresholds and mean average precision / mean average normalized precision at different location error thresholds on testing videos in LaSOT and OTB-99-lang.




(a) Success Plots




(b) Precision Plots




(c) Normalized Precision Plots

Figure 6: Performance of trackers on "NL-consistent LaSOT": one pass evaluation success rate at different intersection over union thresholds and mean average precision / mean average normalized precision at different location error thresholds on testing videos of LaSOT ("NL-consistent LaSOT") with NL annotations verified by crowd sourcing workers. Our model is conditioned on NL and bounding box.

### 4.4. Ablation Studies

In this section, we study the contribution of key aspects of our formulation. We focus on the OTB-99-lang dataset and follow the OPE protocol to enable a more comprehensive comparison with previous methods. The results of this ablation study are summarized in Fig. 8. In Fig. 8a, we observe that disabling relaxed-gating has a relatively minor effect on performance, dropping from 0.61 to 0.6 in terms of success rate. However, the red and dashed-green curves reveal a more complex phenomenon: gating-free tracking appears to yield a higher success rate in the low-

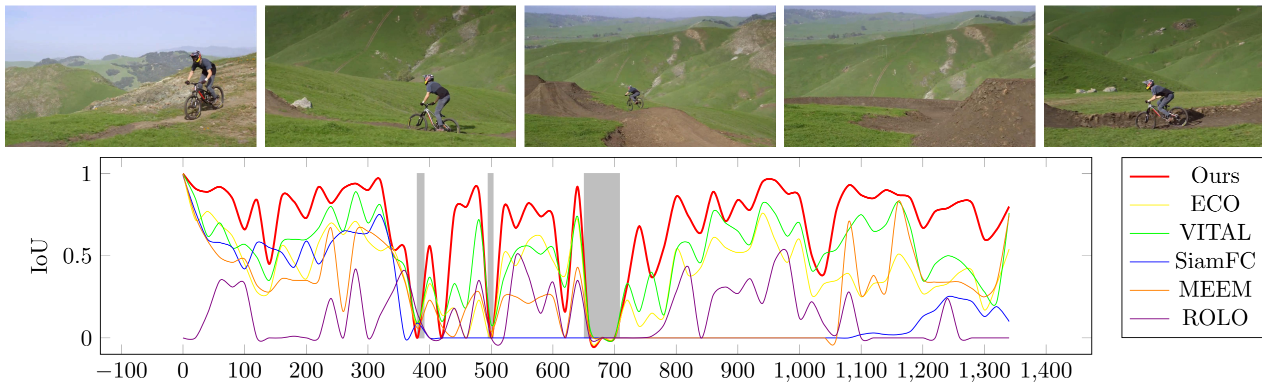
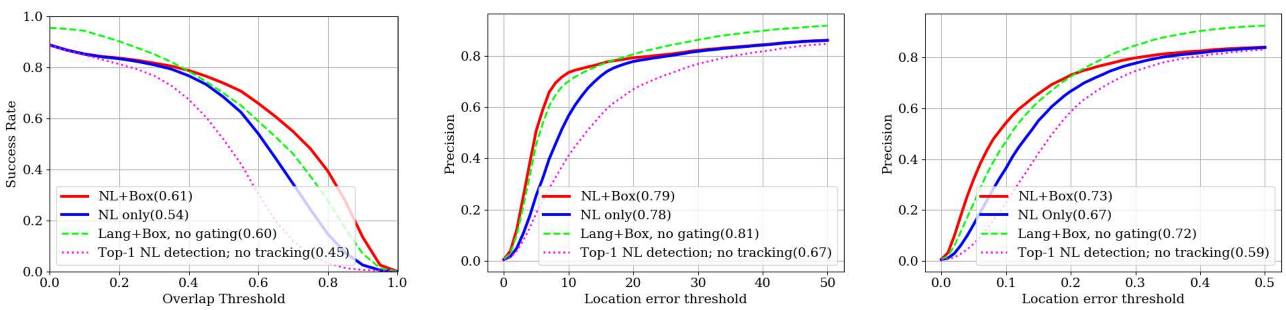



Figure 7: IoU vs. time into tracking on *bicycle-18* from LaSOT [10]. The given NL description is: *black bicycle by a man on the mountain road.* Our model here is evaluated without relaxed gating. Gray temporal intervals indicate full occlusions. Frames on the top are sampled from the video to show how occlusion happened.




(a) Success Plots on OTB

(b) Precision Plots on OTB

(c) Norm. Precision Plots on OTB

Figure 8: Ablation studies on the OTB-99-language dataset using the OPE protocol. Red curve is evaluated on our model conditioned on both NL and bounding box, while the blue curve is evaluated on our model conditioned on NL only. Dashed-green curve is evaluated on our model conditioned on both NL and bounding box but without relaxed gating during tracking. Dashed-magenta curve is evaluated on our NL detection module.

accuracy regime, but under-performs badly as we "require" higher accuracy. This is not surprising: accepting temporally implausible language proposals tends to yield poor tracking results. Conditioning our tracker solely on NL (the blue curve) does not have a substantial effect up to the threshold of 0.5. However, the red curve dominates the blue when using higher thresholds of IoU for evaluation.

Recall from Fig. 5d that our pure-NL tracker is competitive with SiamFC [2]. The remaining curve, dashed-magenta corresponds to picking one bounding box per frame, the one that scores highest by our NL detection stage. Not surprisingly, such a "tracker" does not fare well, underscoring the benefit of our LSTM-based formulation.

In summary, our ablation study supports our argument about the benefit of NL. It's worth noting that this benefit can only be obtained by deriving a reasonable tracking formulation.

## 5. Conclusion

We have presented a novel approach to condition a visual tracker on a natural language description of its target. Unlike prior attempts, we propose a novel track-by-detection route: our formulation employs a deep network to propose image regions that are most likely given the NL description and an RNN that implicitly maintains the history of appearance and dynamics. Experiments on challenging datasets demonstrate that our NL tracker is competitive with the state of the art using standard performance metrics. More importantly, our tracker enjoys better stability in terms of re-acquiring the target after rapid motion or occlusions than conventional trackers. This property supports our claim that NL descriptions provide additional invariants for the target and if exploited correctly, can yield a tracker that is more robust to occlusions.

**Acknowledgement:** This work has been partially supported by Hikvision USA.

## References


[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016. 5

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pages 850–865. Springer, 2016. 1, 2, 5, 6, 8

[3] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004. 2

[4] Cedric De Boom, Steven Van Canneyt, Thomas Demeester, and Bart Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. 2

[5] Eli Brookner. g–h and g–h–k filters. *Tracking and Kalman Filtering Made Easy*, pages 3–63. 2

[6] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4660–4669, 2019. 2

[7] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, Michael Felsberg, et al. Eco: Efficient convolution operators for tracking. In *CVPR*, volume 1, page 3, 2017. 2, 6

[8] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488. Springer, 2016. 2, 5

[9] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011. 5

[10] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. *arXiv preprint arXiv:1809.07845*, 2018. 1, 5, 8

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 1

[12] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 2

[14] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018. 4

[15] Chen Huang, Simon Lucey, and Deva Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 105–114, 2017. 2

[16] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[17] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012. 2

[18] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 2

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 4

[20] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, Nov 2016. 1

[21] Sangkuk Lee, Daesik Kim, Myunggi Lee, Jihye Hwang, and Nojun Kwak. Where to play: Retrieval of video segments using natural-language queries. 2

[22] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019. 1

[23] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018. 1

[24] Shuang Li, Tong Xiao, Hongsheng Li, Bolei Zhou, Dayu Yue, and Xiaogang Wang. Person search with natural language description. *arXiv preprint arXiv:1702.05729*, 2017. 2

[25] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees G. M. Snoek, and Arnold W.M. Smeulders. Tracking by natural language specification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3, 5, 6

[26] Guanghan Ning, Zhi Zhang, Chen Huang, Xiaobo Ren, Haohong Wang, Canhui Cai, and Zhihai He. Spatially

supervised recurrent convolutional neural networks for visual object tracking. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4. IEEE, 2017. 2

[27] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. 2, 3, 4

[28] 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*, pages 91–99, 2015. 2, 4

[29] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2555–2564, 2017. 2

[30] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8990–8999, 2018. 2, 5, 6

[31] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. 2

[32] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *CoRR*, abs/1609.06647, 2016. 2

[33] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *Advances in Neural Information Processing Systems*, pages 809–817, 2013. 2

[34] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006. 1

[35] Jianming Zhang, Shugao Ma, and Stan Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*, pages 188–203. Springer, 2014. 6

[36] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018. 1