Distractor-Aware Fast Tracking via Dynamic Convolutions and MOT Philosophy

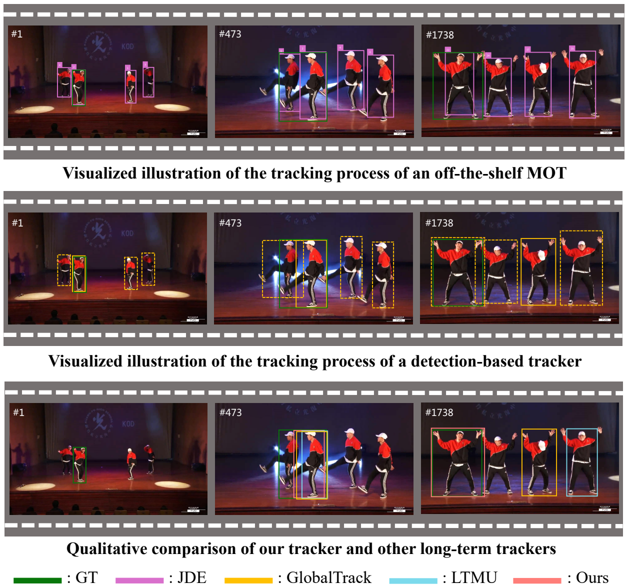
Zikai Zhang[1,2], Bineng Zhong[1,*], Shengping Zhang[3,4], Zhenjun Tang[1], Xin Liu[5], Zhaoxiang Zhang[6]
[1]Guangxi Key Lab of Multi-Source Information Mining & Security, Guangxi Normal University,
[2]Department of Computer Science and Technology, Huaqiao University,
[3]Harbin Institute of Technology, [4]Peng Cheng Laboratory, [5]Beijing Seetatech Technology,
[6]Institute of Automation, CAS & University of Chinese Academy of Sciences
& Centre for Artificial Intelligence and Robotics, HKISI_CAS
zikaizhang@hqu.edu.cn, bnzhong@gxnu.edu.cn, s.zhang@hit.edu.cn
tangzj230@163.com, xin.liu@seetatech.com, zhaoxiang.zhang@ia.ac.cn

Abstract

A practical long-term tracker typically contains three key properties, i.e. *an efficient model design, an effective global re-detection strategy and a robust distractor awareness mechanism. However, most state-of-the-art long-term trackers* (e.g., *Pseudo and re-detecting based ones) do not take all three key properties into account and therefore may either be time-consuming or drift to distractors. To address the issues, we propose a two-task tracking framework (named* **DMTrack***), which utilizes two core components* (i.e., *one-shot detection and re-identification (re-id) association) to achieve distractor-aware fast tracking via* **D***ynamic convolutions (d-convs) and* **M***ultiple object tracking (MOT) philosophy. To achieve precise and fast global detection, we construct a lightweight one-shot detector using a novel dynamic convolutions generation method, which provides a unified and more flexible way for fusing target information into the search field. To distinguish the target from distractors, we resort to the philosophy of MOT to reason distractors explicitly by maintaining all potential similarities' tracklets. Benefited from the strength of high recall detection and explicit object association, our tracker achieves state-of-the-art performance on the LaSOT, OxUvA, TLP, VOT2018LT and VOT2019LT benchmarks and runs in real-time (3x faster than comparisons)*[1].

1. Introduction

Visual object tracking has drawn great attention to large-scale long-term tracking because of its great potential in real-world applications. The main difference between long-term and short-term trackers is that the former has to deal with the cases in which the target disappears and reappears frequently. Generally, long-term tracking sequences [8, 26, 31] last for hundreds and thousands frames, which usu-



Visualized illustration of the tracking process of an off-the-shelf MOT

Visualized illustration of the tracking process of a detection-based tracker

Qualitative comparison of our tracker and other long-term trackers

Figure 1. Visualization of long-term tracking results on *person-5* from LaSOT [8]. "GT" means ground truth. "GlobalTrack [12]" and "LTMU [4]" are two strong long-term trackers. "JDE [34]" is a multi-object tracker. In the first line, we show long-term tracking results from the off-the-shelf MOT model[34]. The object ids are in the upper left corner of the bounding boxes. In the second line, we show top-4 classification results from a detection-based tracker[12]. The solid lines show the top-1 predictions. In the third line, we compare our DMTrack with state-of-the-art comparison, and present that distractor awareness is vital to visual object trackers. Better viewed in color with zoom-in.

[*] Corresponding author.
[1] The code will be available at https://github.com/hqucv/dmtrack

ally contain challenges such as appearance change, long-duration disappearance and intra-class distractors. Therefore, long-term trackers should have the abilities of re-detecting objects effectively and distinguish the target from similar distractors (As shown in Figure 1).

Recently, a large number of long-term trackers have been proposed [4, 37, 41, 42]. Lukežič *et al*. [20] group long-term trackers into two categorizations: Pseudo long-term tracker (LT_0) and re-detecting long-term tracker (LT_1). LT_0 applies some short-term trackers [6, 45, 43] to the long-term tracking task straightforwardly by simply using the classification score to distinguish the target from its background. However, these trackers are prone to drift to distractors due to appearance confusion. LT_1 (e.g., SiamDW_LT [42], MBMD [41], SPLT [37], LTMU [4]) uses a re-detection strategy to recover from tracking failure. However, these trackers require a sophisticated design for interaction between local trackers and global detectors. Recently, Huang *et al*. [12] propose a global instance search (GIS) based tracker using a two-stage detector without motion constraint. Under the one-shot detection scheme, long-term tracking is simplified because the switch strategy that used for balancing the local and global modules is no longer needed. However, the heavy computing burdens and unstable performance caused by global detection make GIS-based methods improper for real-world applications.

To address the above issues, we propose a two-task tracking framework, which consists of a lightweight detection model and an explicit object association method. For the first task, we reach back to the correlation methods between the template and the search field in tracking and unify these methods into a dynamic convolutions (d-convs) generation paradigm [38]. Given powerful dynamic convolutions, we can embed target information into a one-stage anchor-free detection model with multiple kernel designs and integration layers while requiring less computation. For the second task, we resort to the philosophy of multiple object tracking (MOT). Specifically, we introduce a novel re-id embedding into the above detection model by jointly learning the two tasks. Benefiting from the discriminative re-id features, our tracker achieves favorable performance with a compact association strategy. The two tasks are implemented in the MOT framework which reasons distractors explicitly by maintaining all potential similarities tracklets. Experiments show that our tracker achieves state-of-the-art performance on the five long-term benchmarks and runs 3x faster than comparisons.

Our main contributions can be summarized as follows,

- We propose a two-task long-term tracking framework, which contains a lightweight detector and an explicit object association. By implementing the two tasks in the MOT framework, our tracker obtains a fast inference speed and can distinguish the target from the distractors.
- To build a high-efficiency detector, we present a novel dynamic convolutions generation method. To avoiding drifting to distractors, we learn a discriminative re-id embedding to achieve effective tracklet association.
- Our approach achieves state-of-the-art results on five long-term tracking benchmarks and runs in real-time, which shows that the proposed method can be a more practical baseline for GIS-based trackers.

2. Related Work

2.1. Deep Long-Term Visual Tracking

Deep learning-based models for long-term tracking have shown their great capability [9, 31, 37, 41, 47]. Recent top-ranked long-term trackers follow the local tracker and global re-detector schemes. MBMD [41] combines regression and verification modules to the tracking framework, and use a sliding window strategy in image level for re-detecting. SPLT [37] uses a skimming module to speed up the re-detect processing by skipping the certain regions. However, there is a tough problem in the local-global paradigm: *when to switch between the local tracker and the global re-detector?* However, some methods followed the pure re-detection paradigm, Huang *et al*. [12] proposed to track in global search scheme by introducing a two-stage anchor-based detection framework in tracking task. Voigtlaender *et al*. [32] further use a cascade detection head for precise results. However, the computing burdens are heavy in these methods. In this paper, we develop an efficient detection model under one-stage anchor-free paradigm, which gets a favorable balance between speed and accuracy.

2.2. Distractor Problem in Visual Tracking

The ability of dealing with similar objects is important for long-term tracker. However, distractor problem is ill-posed due to the dynamic interaction of target and distractors. Zhu *et al*. [47] propose an incremental learning method for online distractor suppression. Voigtlaender *et al*. [32] implement a hard example mining to guide model learning and use a dynamic programming algorithm to consistently suppress potential distractors. Nevertheless, these methods are burdensome for accurate and efficient tracking. In this work, we address distractor problem by explicitly tracking all the potential objects in the MOT framework. Inspired by the joint learning methods of MOT [34, 40, 46], we design a compact two-task tracking framework with one-shot detection and re-id association that runs in real-time.

2.3. Correlation Methods for Visual Tracking

Correlation operations that used for fusing the template information and the search field are seldom discussed.

Bertinetto *et al*. [1] use a simple cross-correlation operation to generate a similarity scoring map. Li *et al*. [17] proposed a depthwise cross-correlation to reduce the computational cost. Huang *et al*. [12] use the Hadamard production to encode correlation information. However, these methods are restricted to fixed model structures. Contrast to fixed convolutional layer designs, d-convs are dynamically generated by using some conditional information. Dynamic filter network [13] and CondConv [38] explore the power of d-convs for increasing the capacity of the classification model. CondInst [29] and SOLOv2 [33] use conditional convolution to embed position information into the mask branch for boosting the segmentation performance. Here, we utilize a dynamic convolutions generation method for feature correlation.

3. Method

In this section, we unveil the power of GIS-based trackers [11, 12, 32] by designing a lightweight detector and a re-id embedding with capable association strategy. As shown in Figure 3, our DMTrack consists of a group of dynamic convolution controllers, an efficient one-shot detection branch and a re-id embedding.

3.1. Motivation

What is the strength of global search scheme for long-term tracking? How can we achieve a more stable global tracker? We dive into GIS-based methods, and answer these two questions by designing experiments to analysis the capabilities of modern trackers. For the first question, we show GIS-based methods have a high-quality proposal generating ability. And for the second question, we demonstrate the importance of the association ability by using an off-the-shelf MOT model to evaluate on a single object tracking benchmark. These two experiments are meant to show important factors that we must take into account when building a practical GIS-based tracker.

Proposal Generator. High-quality proposals are important for a tracking system. Though local search-based trackers achieve favorable performance in short-term scenes, most of them degenerate in large-scale long-term benchmarks [8, 22, 26, 31]. Therefore, a global re-detector becomes a core component for long-term tracking. Actually, we can treat a global re-detector as a proposal generator. And a high-recall generator with only a few candidates will be beneficial to later tracking stages.

We experimentally evaluate the upper bounds of local and global trackers. Following popular protocol in OTB-2015 [35], we perform the One-Pass Evaluation (OPE) and measure the best success score of top-K candidates of a local generator and a global generator. Here, we evaluate RT-MDNet [14] as the local generator (marked as “RT-MDNet*”) and GlobalTrack [12] as the global generator (marked as “GlobalTrack*”). For RT-MDNet, the tracker makes a gaussian sampling based on the position of the last frame prediction, we simply choose the top-k candidates by classification scores, then we determine the bounding box that has the highest intersection over Union score (IoU) as our output. For GlobalTrack, we just follow the identical strategy. Specifically, we implement the same classifier in these generators to control the variable. In the experiment, we set the candidate number K with 2, 3, 5, 15, 50, 100 and test on the LaSOT [8]. As shown in Figure 2, GlobalTrack* exceed RT-MDNet* by a large margin. Even with only two candidates, the global-based generator achieves a success score of 61.3%, which is comparable with the local-based generator with the top-50 score, and outperforms state-of-the-art long-term tracker LTMU [4]. However, we can see that GIS-based method is time-consuming due to the heavy model design for global search (as shown in Figure 2). Therefore, one of the important factors that contribute to tracker's performance is a good balance between the precision and the efficiency.

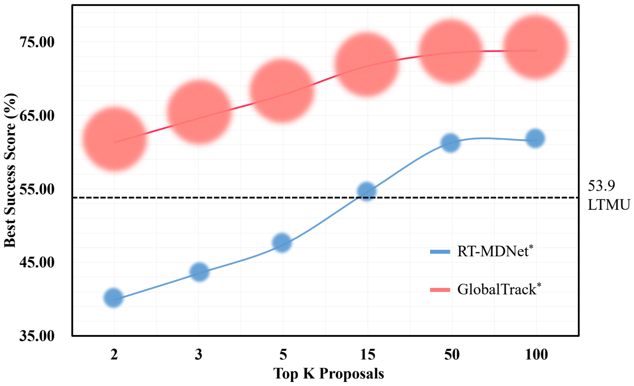


Figure 2. Visualization of best success scores on LaSOT [8] and speeds from the local and global generator. The circle diameter means the relative time cost in inference stage.

Off-the-Shelf MOT for Long-Term Tracking. The above experiments have shown the capability of GIS-based trackers. In spite of the high proposal recall, the discrimination of these trackers is unsatisfactory. Intra-class objects reasoning is crucial for global search scheme. Here, we design an enlightening experiment to evaluate the long-term tracking performance of an off-the-shelf MOT model. Specifically, we choose sequences in long-term tracking benchmarks [8] that contains categories of *person* (match with MOT pre-trained model). During inference stage, we maintain the object id that has the max IoU between the detections and the annotation in the first frame. In the subsequent frames, we just choose the prediction with the same id as our tracking result. In Figure 1, we present the tracking results of JDE [34]. As we can see, MOT method keeps a robust and accurate tracking for the first hundred frames,

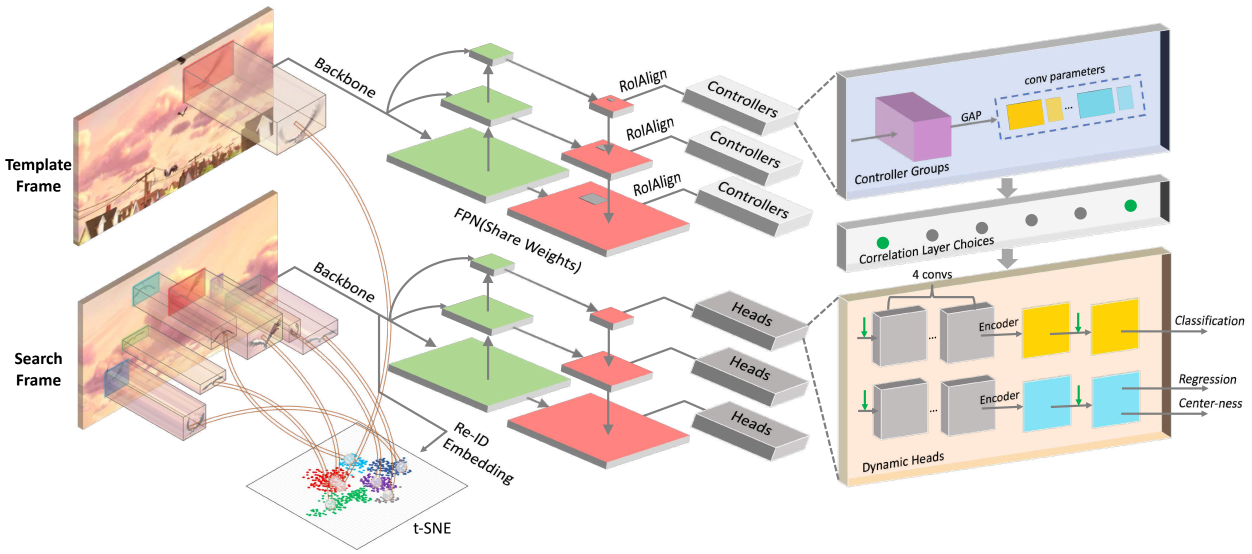


Figure 3. The overall architecture of our model. The framework consists of three main components: a template branch for dynamic convolutions generation, a search branch for efficient one-shot detection and a re-id embedding for object association.

which is surprising because the model is never trained on single object tracking dataset. Therefore, besides high-recall proposal generator, object association is also an important factor that carries a big weight in final tracking performance.

Motivated by the above trials, we design a GIS-based tracker that considers computation cost and distractor awareness. In the next part, we make an overview on our DMTrack and demonstrate its core components.

3.2. Overall Architecture

Given the template image $I^t \in \mathbb{R}^{H \times W \times 3}$ and search image $I^s \in \mathbb{R}^{H \times W \times 3}$, our tracking algorithm search the target in successive frames with only first frame annotation. In this work, we develop a GIS-based tracker that uses an anchor-free detection model as the base detection model. In order to build a class-agnostic detector, we introduce the dynamic convolutions controller to generate convolution parameters conditioned on the target information. Further, we jointly learn the detection task with a re-id embedding model for the detection and the association stages. The total framework is shown in Figure 3.

We design our detection model under anchor-free paradigm [30]. Being benefited from compact model design and simplified parameter settings, the detection branch obtains a satisfactory inference speed. As shown in Figure 3, we use DLA-34 [39] as our model backbone and feature pyramid networks (FPN [18]) as our model neck. We aggregate feature maps by FPN and use multiple-scale features from three levels P3, P4, P5, the strides s of features are 8, 16, 32, respectively. The backbone and neck are shared in both template and search branch.

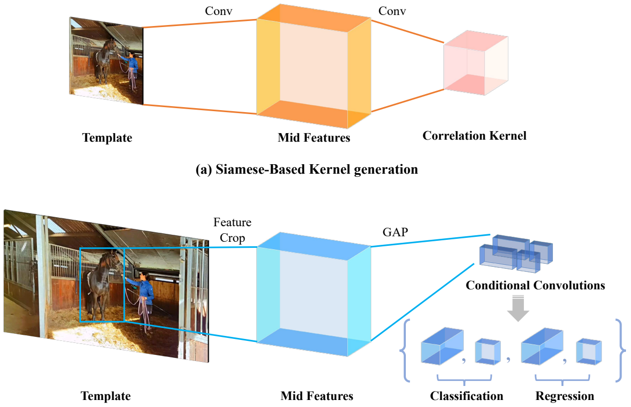
In template branch, we use an efficient feature align method [10] to crop target features. And then a group of controllers that use these features to generate convolution parameters for d-convs. In search branch, following the stacking designs of modern detection methods [19, 30], we embed our d-convs in specific layers to filter the useful features. In re-id embedding branch, we design to generate N-dimensions re-id features for each point in stride-4 feature map. N is set to be 128 in our model. With the discriminative re-id feature, we obtain smooth tracking trajectories.

3.3. One-shot Detection and Embedding Learning

We follow the common practice of GIS-based tracker to train a class-agnostic detector with d-convs. Firstly, by using a full convolutional network, each location $(x_i^P, y_j^P)(P = 3, 4, 5)$ on the feature map of different FPN levels can be mapped back onto the original image as

$$(x_{i'}^{ori}, y_{j'}^{ori}) = (\left\lfloor \frac{s^P}{2} \right\rfloor + x_i^P s^P, \left\lfloor \frac{s^P}{2} \right\rfloor + y_j^P s^P) \tag{1}$$

where i and j indicate the x, y-coordinates on the feature map, ori indicates original input image. Then we define a center region box on original image as the sampling box $(c_i^{ori} - rs^P, c_j^{ori} - rs^P, c_i^{ori} + rs^P, c_j^{ori} + rs^P)$, where (c_i^{ori}, c_j^{ori}) denotes the annotation center of the target, r is a scale parameter being 1.5, which is the same as the default setting [30]. According to the former definition, we define the location on FPN's feature that can be mapped onto the center region box on original input image as a positive sample $\sigma(= 1)$, otherwise a negative sample. Note that



(a) Siamese-Based Kernel generation

(b) Dynamic Generated Conditional Convolutions

Figure 4. Comparison of different kernel generation methods. (a) Siamese-based method generates a large kernel by extracting feature from a coarse template image which includes many noises. (b) With aligned feature cropping [10], convolutions were generated by a global average pooling layer, which are more flexible and effective.

we train the regression head and center-ness head only on positive samples. For re-id branch, we treat it as a classification task. As there are no multiple objects annotations in single object tracking dataset, we introduce MOT datasets [7, 25, 36, 44] to train our re-id branch alternatively. Finally, our model predicts a 2-D vector $\hat{\sigma}$ for classification, a 4-D vector $\hat{\tau} = (\hat{l}, \hat{t}, \hat{r}, \hat{b})$ for bounding box regression, where $(\hat{l}, \hat{t}, \hat{r}, \hat{b})$ indicates the distances from the box center to four sides, a center-ness score $\hat{\varphi}$ for classification regularization and a 128-D re-id embedding feature $\hat{\psi}$.

Controller Heads. How to extract abundant target features is an important problem for visual tracking. Nevertheless, this problem is seldom discussed. In Siamese-based methods [1, 3, 17], the template branch uses a coarse cropping to extract features, which is not appropriate for similar matching due to the noises it involved. Furthermore, the size and the type of the Siamese kernel are fixed for the final activation map which is difficult for model reconstruction. In the dynamic generating method, firstly, we extract the target information from head layers using a feature cropping technology[10]. Then with a 1×1 $conv$ encoder to adjust the feature channels $\mathcal{C}^g$ (as shown in Equation 2) to adapt to the required parameter numbers as shown in Equation 3. Finally, we use a global average pooling layer to generate $\mathcal{C}^g - D$ vectors for filter parameters of classification and regression heads.

$$\mathcal{C}^g = \sum_{u=1}^{p} \mathcal{PN}(conv^u_{cls}) + \sum_{v=1}^{q} \mathcal{PN}(conv^v_{reg}) \tag{2}$$

$$\begin{aligned} \mathcal{PN}(conv^u_{cls}) &= (C^u \times K^u_w \times K^u_h + 1) \times C^{u+1} \\ \mathcal{PN}(conv^v_{reg}) &= (C^v \times K^v_w \times K^v_h + 1) \times C^{v+1} \end{aligned} \tag{3}$$

where p, q denote layer numbers counted after model's neck (from 1 to 6). $\mathcal{PN}$ means *parameter numbers* of the d-convs, e.g. , in u layer of classification branch, the amount of parameters consist of weights and bias. Specifically, the weight's parameters can be a multiplication of input feature map's channel C^u, kernel width K^u_w, kernel height K^u_h and kernel number C^{u+1}.

Detection Heads with Dynamic Convolutions. The detection head contains four components: classification, regression, center-ness and dynamic convolutions. In our model structure, there are four convolutions after neck, an encoder to reduce the channels for efficient computation and prediction layers for each head. We first define numbers of feature layers for classification and regression heads which from 1 to p and q, respectively. Then we define groups of layers $\{u_i \mid 0 \leqslant i \leq p, i \in \mathbb{N}^*\}$ and $\{v_j \mid 0 \leqslant j \leq q, j \in \mathbb{N}^*\}$ to insert our d-convs. By integrating target information to detection head with d-convs, we predict the target and directly regress bounding box at each location on feature maps. Following FCOS [30], we also predict the center-ness scores associate with regression branch for further robustness.

Re-id Embedding. We jointly learn detection task and re-id embedding in order to distinguish similar objects. Here we use a convolutional layer on the low level of backbone features to extract re-id embedding features with stride 4. Each re-id feature $E(x, y) \in \mathbb{R}^{C^E}$ in location (x, y) present the object whose center annotation that closest to it , C^E is the dimension of the embedding feature and set to 128 in our settings.

3.4. Loss Function

We model the learning task of our framework as a multi-task problem. There are two learning objectives in our full pipeline: detection and re-id. For the detection part, we have three loss functions for classification $\mathcal{L}_\alpha$, regression $\mathcal{L}_\beta$ and center-ness $\mathcal{L}_\gamma$ as following

$$\mathcal{L}_\alpha(\sigma_{i,j}) = \sum_{i,j} \mathcal{L}_{focal}(\hat{\sigma}_{i,j}, \sigma_{i,j}) \tag{4}$$

$$\mathcal{L}_\beta(\tau_{i,j}) = \begin{cases} \sum_{i,j} \mathcal{L}_{IoU}(\hat{\tau}_{i,j}, \tau_{i,j}) & \sigma_{i,j} = 1 \\ 0 & otherwise \end{cases} \tag{5}$$

$$\mathcal{L}_\gamma(\varphi_{i,j}) = \begin{cases} \sum_{i,j} \mathcal{L}_{BCE}(\hat{\varphi}_{i,j}, \varphi_{i,j}) & \sigma_{i,j} = 1 \\ 0 & otherwise \end{cases} \tag{6}$$

The detection loss of multiple scales and heads can be summarized as

$$\mathcal{L}_{\det} = \sum_{m=3,4,5} \sum_{n=\alpha,\beta,\gamma} \lambda^m_n \mathcal{L}^m_n \tag{7}$$

where λ^m_n are loss weights to balance these loss functions. Further, we follow the cross-query loss in [12] to improve the discriminating ability of our method as following

CC	DW	HP	DC	Layers	Top-1	PC	AC	Re-id	Success↑	Precision↑	FPS↑	GPU Days↓
✓				1+6	✓	-	-	-	49.8	50.0	27	15
	✓			1+6	✓	-	-	-	51.4	52.4	32	9.5
		✓		1+6	✓	-	-	-	51.1	52.5	32	9.5
			✓	1+6	✓	-	-	-	53.0	54.2	32	10
-	-	-	✓	6	✓	-	-	-	48.7	49.2	34	9
-	-	-	✓	$[1 \rightarrow 6]$	✓	-	-	-	53.2	54.3	28	19
-	-	-	✓	1+6		✓			46.9	42.4	32	10
-	-	-	✓	1+6			✓		55.2	55.9	16	10
-	-	-	✓	1+6				✓	57.4	58.0	31	11

Table 1. Ablation studies of different model designs and object association strategies that influences the model's capacity. We also evaluate the time cost for training and inference on a Titan Xp.

$$\mathcal{L}_{\text{det}}^{'} = \frac{1}{I}\sum_{i=1}^{I}\mathcal{L}_{\text{det}} \tag{8}$$

where I indicates the template-search pairs for a pair of images, which means we calculate average loss over different targets in one search image.

For the re-id embedding part, we treat object identity as a classification problem and use loss function like in [40] for the model training

$$\mathcal{L}_{reid} = \sum_{i=1}^{M}\sum_{m=1}^{J}\mathcal{L}_{softmax} \tag{9}$$

where J is the number of classes, M is the number of objects. And we use strategy in [34] to balance the detection and re-id loss.

3.5. Online Tracking

Network Inference. The inference of our model is straightforward. We initialize the dynamic convolutions using the first frame annotation and keep the re-id feature of the target. Then in subsequent frames, we use the generated kernels to convolve the feature maps in multiple layers. Finally, we take the top-k candidates ordered by the classification score and use a variant non-maximum suppression (NMS) strategy to provide a group of interests.

Online Box Linking. We use three clues for box linking as in [40]: appearance information (*i.e.* re-id features), position information (*i.e.* IoU between adjacent frames) and motion information (*i.e.* Kalman Filter). With these abundant clues, we get smooth box linking with simple Hungarian algorithm [16].

4. Experiments

4.1. Implementation Details

Parameters. We use light version of FCOS [30] with DLA-34 [39] backbone as our base model for one-shot detection.

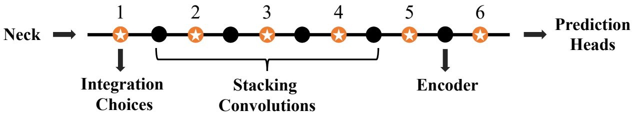


Figure 5. Numbering the integration layer choices for one-shot detection head.

The feature channels are 256 for four stack convolutions and 32 for the encoder behind these. In the template branch, we use RoIAlign [10] with output feature size of 7. Then we use a group of k controllers with global average pooling to generate dynamic convolutions (we set k to 4 as shown in Section 4.2). In our model, we simply generate 1×1 convolutions. In the search branch, we embed d-convs behind neck layer and stack convolutions for both classification head and regression head. In re-id embedding, we use a convolution layer on top of the backbone features with 128 channels, the feature map size is a quarter of size of input image.

Training. We use the same training data as in [12, 40] and use the multi-scale data augmentation by sampling shorter size of input image from 256 to 608 with interval 32. Our model is trained with stochastic gradient descent (SGD) with a starting learning rate of 1×10^{-3}. We use 360K training iterations and decreased by 10 at iteration 300K and 340K respectively.

4.2. Ablation Study

In this section, we conduct ablation analysis to evaluate different components of our tracker using the LaSOT [8] benchmark. The image size for inference is set to 735×512 for all testing.

Effectiveness of Correlation Method. We make quantitative analysis to compare our dynamic convolutions (DC) generation method with other correlation methods. We denote the cross correlation [1] by CC, depthwise cross correlation [17] by DW, Hadamard production [12] by HP. As shown in Table 1, with similar inference time, we show

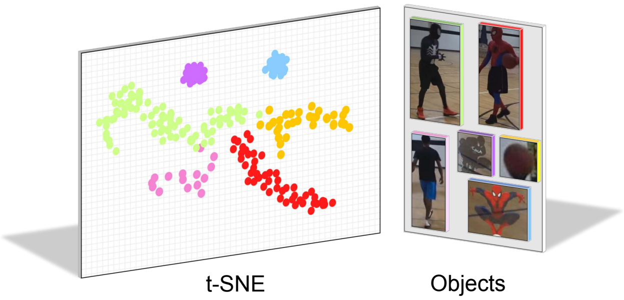


Figure 6. We show the effectiveness of re-id embedding by using t-SNE [24] to visualize the distance between the features of different objects. The features of the same object are shown by the same color.

that d-convs based correlation method are more powerful to model the template information and embed it into search field. This fine-grained feature learning results in strong template correlation. Other methods (i.e. Siamese-based and modulation-based) are special cases of D-Conv. With compact convolutions, we provide a non-trivial solution to unify previous methods, and urge further research on this problem.

Integration Layer Choices. We experimentally evaluate the influences of integration layer choice, a key factor for correlation capacity. First, we show the sketch of head structures in Figure 5, there are four stacking convolutions and one encoder, therefore the permutation and combination of six candidate integration layers can result in hundreds choices. Here, we show the relation between the performance and training cost in Table 1. From the table, we can see that with only the high-layer integration (line 5), tracker gets degenerate results. However, with dense connections in stacking layers (line 6), the performance does not boost significantly, but the training cost can be unbearable. In our final model, we integrate d-convs with the *1+6* layers for more practical.

Effectiveness of objects association with re-id embedding. GIS-based trackers can potentially track all interested objects. These trackers treat the object that have the Top-1 classification score as the target. However, distractor problem leads the tracking performance to deterioration because trackers do not use any constraint. We implement two heuristic constraints to compare with our association method. The first one uses position constraint (denoted by PC). In this constraint, we simply choose the object which is closest to last frame prediction among top 5 candidates. The second one uses appearance constraint (denoted by AC). In this constraint, we use an extra classifier [2] with online update to choose the final target. As shown in Figure 1, our association strategy (denoted by Re-id) outperforms two heuristic methods by a large margin both in precision and computational costs. With only the position or appearance constraint, tracker unable to deal with high-frequency disappearance. However, with an explicit multiple-object association, our tracker is more robust to these real-world challenges. Besides, we use t-SNE [24] technology to show re-id features for different objects. As shown in Figure 6, we use *person-6* sequence from LaSOT [8] and label six instances in these frames (include *person*, *basketball* and *watermark* on the videos). We show that the re-id embedding can differentiate the inter-class objects and the intra-class objects.

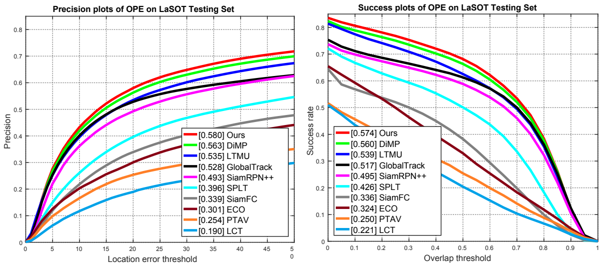


Figure 7. Plots of ours and state-of-the-art trackers on the test set of LaSOT [8]. Better viewed in color with zoom-in.

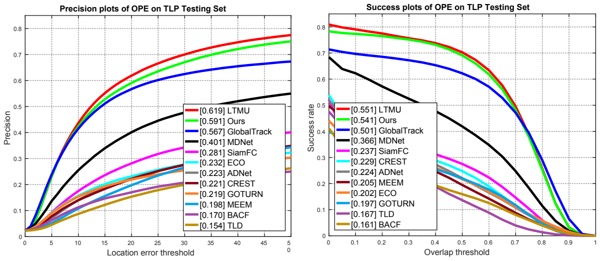


Figure 8. Plots of Ours and state-of-the-art trackers on the test set of TLP [26]. Better viewed in color with zoom-in.

4.3. Comparison with the state-of-the-art

LaSOT. The LaSOT benchmark [8] is a large-scale modern tracking dataset that contains 1400 long videos (with an average of 2500 frames). In this work, we follow the protocol II defined by official evaluation toolkit and conduct one-pass evaluation with success and precision scores to evaluate our tracker. Compared to nine SOTA methods [1, 2, 4, 5, 9, 12, 17, 23, 37][2], our approach achieves the best results among all competing methods. As shown in Figure 7, our tracker achieves the best results among all competing methods. Besides that, we maintain a fast inference speed with compact model design, which shows the practicability of our approach.

OxUvA. The OxUvA [31] is a long-term tracking dataset in the wild. The dataset consists of 366 object tracks which

[2] We use the raw result provided by official evaluation toolkit in their website.

(%)	MaxGM	TPR	TNR	FPS
ECO-HC [5]	31.4	39.5	0.0	-
MDNet [27]	34.3	47.2	0.0	1
LCT [23]	39.6	29.2	53.7	20
TLD [15]	43.1	20.8	89.5	23
MBMD [41]	54.5	60.9	48.5	3
GlobalTrack[12]	60.3	57.4	63.3	10
SPLT [37]	62.2	49.9	77.6	27
Siam R-CNN [32]	72.3	70.1	74.5	4.7
Ours	68.8	68.6	69.4	31

Table 2. State-of-the-art comparison on the test set of OxUvA [31] in terms of MaxGM, TPR and TNR. The best three results are shown in red, blue and green colors, respectively.

VOT2018LT				VOT2019LT			
Tracker	**F-score**	**Pr**	**Re**	**Tracker**	**F-score**	**Pr**	**Re**
PTAVplus	0.481	0.595	0.404	FuCoLoT	0.411	0.507	0.346
SYT	0.509	0.520	0.499	ASINT	0.505	0.517	0.494
LTSINT	0.536	0.566	0.510	CooSiam	0.508	0.482	0.537
MMLT	0.546	0.574	0.521	SiamRPNsLT	0.556	0.749	0.443
DaSiam_LT	0.607	0.627	0.588	mbdet	0.567	0.609	0.530
MBMD	0.610	0.634	0.588	SiamDW_LT	0.665	0.697	0.636
SPLT	0.616	0.633	0.600	CLGS	0.674	0.739	0.619
SiamRPN++	0.629	0.649	0.609	LT_DSE	0.695	0.715	0.677
Ours	0.683	0.687	0.655	**Ours**	0.687	0.690	0.662

Table 3. State-of-the-art comparison on the VOT2018LT [21] and VOT2019LT [22] benchmarks in terms of F-score, Pr and Re. The best three results are shown in red, blue and green colors.

are chosen from YTBB [28] and labeled at 1Hz frequency. According to the [31], OxUvA is divided into two subset: *dev* and *test*. The test subset contains 166 tracks and each of these lasts for average 2.4 minutes. The evaluation criteria is quite different from short-term benchmarks [35, 22], we introduce them briefly as following. The true positive rate (**TPR** calculate the fraction of present objects that are predicted present and precisely. The true negative rate (**TNR**) gives the fraction of absent objects that are determined to disappear. The **MaxGM** provides more convinced measurement to show the trackers performances and is defined as

$$\mathbf{MaxGM} = \underset{0\leq p\leq 1}{max} \sqrt{((1-p)\cdot\mathbf{TPR})((1-p)\cdot\mathbf{TNR}+p)} \tag{10}$$

We compare our method with eight competing approaches using the open challenge illustrated in [31]. In this challenge, trackers can use any public dataset as the training data expect for the YTBB [28] validation set. As we can see in Table 2, our method achieves comparable performance to sophisticated designed long-term trackers that have heavy computation. However, our method runs in real-time, which is practical for applications.

TLP. The TLP is a long video dataset for object tracking. The dataset including 50 long videos of 676K frames (over 400 minutes). We follow the OPE evaluation that used in [35] and compare our tracker with other trackers. As shown in Figure 8, our tracker outperforms another GIS-based tracker[12] by a large margin and gets a comparable performance to the best tracker[4] in this benchmark. Compared with another GIS-based tracker GlobalTrack [12], our model get a good balance between precision and recall.

VOT2018LT. We compare our tracker with other state-of-the art tracking algorithms on VOT2018LT benchmark [21]. In this dataset, there are 35 long videos with 146K frames in total. The challengs in these sequences are varied, including long-term target disappearances and severe occlusion, which require trackers to be more robust. The evaluation criterion of VOT2018LT dataset includes tracking precision (**Pr**), tracking recall (**Re**) and tracking **F-score**. We report the tracking performance of our tracker and other competing ones in Table 3. As we see, our tracker achieves an absolute gain of 5% in terms of F-score. The results demonstrate the strong performance of our approach in long-term tracking scenarios.

VOT2019LT. The VOT2019LT benchmark [22] is an extension of the 2018 version [21] that contains 50 challenging sequences. Each video contains 10 long-range disappearances on average. The evaluation protocol is similar to that in VOT2018LT [21]. Table 3 show that our model get a promising result compared to the well-designed long-term trackers for the competition. The tracking results demonstrate the advantage of GIS-based paradigm.

5. Conclusions

In this work, we propose a new long-term tracking paradigm which consists of one-shot detection and object association. To achieve an efficient detection model, we design a novel dynamic convolutions generation method for flexible feature correlation. Further, in order to distinguish the target from distractors, we present a compact object association strategy with discriminative re-id embedding. Numerous experiments on five long-term tracking benchmarks verify the performance of the proposed approach. Potentiated by its efficiency, we believe that the proposed framework can be performed as a new baseline for further studies.

Acknowledgments.

This work was supported by the National Natural Science Foundation of China (No. 61972167, 61802135, 61872112), the Project of Guangxi Science and Technology (No. 2020AC19194), the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, the Guangxi Talent Highland Project of Big Data Intelligence and Application, and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (No. 202000012).

References

[1] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops (2)*, volume 9914 of *Lecture Notes in Computer Science*, pages 850–865, 2016. 3, 5, 6, 7

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pages 6181–6190. IEEE, 2019. 7

[3] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, pages 6667–6676. IEEE, 2020. 5

[4] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In *CVPR*, pages 6297–6306. IEEE, 2020. 1, 2, 3, 7, 8

[5] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, pages 6931–6939. IEEE Computer Society, 2017. 7, 8

[6] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, pages 4310–4318. IEEE Computer Society, 2015. 2

[7] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *CVPR*, pages 304–311. IEEE Computer Society, 2009. 5

[8] 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. In *CVPR*, pages 5374–5383. Computer Vision Foundation / IEEE, 2019. 1, 3, 6, 7

[9] Heng Fan and Haibin Ling. Parallel tracking and verifying. *IEEE Trans. Image Process.*, 28(8):4130–4144, 2019. 2, 7

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988. IEEE Computer Society, 2017. 4, 5, 6

[11] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Bridging the gap between detection and tracking: A unified approach. In *ICCV*, pages 3998–4008. IEEE, 2019. 3

[12] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Globaltrack: A simple and strong baseline for long-term tracking. In *AAAI*, pages 11037–11044. AAAI Press, 2020. 1, 2, 3, 5, 6, 7, 8

[13] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NIPS*, pages 667–675, 2016. 3

[14] Ilchae Jung, Jeany Son, Mooyeol Baek, and Bohyung Han. Real-time mdnet. In *ECCV (4)*, volume 11208 of *Lecture Notes in Computer Science*, pages 89–104. Springer, 2018. 3

[15] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1409–1422, 2012. 8

[16] Harold W. Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming*, pages 29–47. Springer, 2010. 6

[17] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4282–4291. Computer Vision Foundation / IEEE, 2019. 3, 5, 6, 7

[18] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944. IEEE Computer Society, 2017. 4

[19] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007. IEEE Computer Society, 2017. 4

[20] Alan Lukežič, Luka Čehovin Zajc, Tomáš Vojíř, Jiří Matas, and Matej Kristan. Performance evaluation methodology for long-term single-object tracking. *IEEE Transactions on Cybernetics*, 2020. 2

[21] Kristan M, Matas J, and et al. Leonardis A. The sixth visual object tracking vot2018 challenge results. In *ECCV Workshops (1)*, volume 11129 of *Lecture Notes in Computer Science*, pages 3–53. Springer, 2018. 8

[22] Kristan M, Matas J, and et al. Leonardis A. The seventh visual object tracking VOT2019 challenge results. In *ICCV Workshops*, pages 2206–2241. IEEE, 2019. 3, 8

[23] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang. Long-term correlation tracking. In *CVPR*, pages 5388–5396. IEEE Computer Society, 2015. 7, 8

[24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 7

[25] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *CoRR*, abs/1603.00831, 2016. 5

[26] Abhinav Moudgil and Vineet Gandhi. Long-term visual object tracking benchmark. In *ACCV (2)*, volume 11362 of *Lecture Notes in Computer Science*, pages 629–645. Springer, 2018. 1, 3, 7

[27] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302. IEEE Computer Society, 2016. 8

[28] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *CVPR*, pages 7464–7473. IEEE Computer Society, 2017. 8

[29] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. *CoRR*, abs/2003.05664, 2020. 3

[30] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: A simple and strong anchor-free object detector. *CoRR*, abs/2006.09214, 2020. 4, 5, 6

[31] Jack Valmadre, Luca Bertinetto, João F. Henriques, Ran Tao, Andrea Vedaldi, Arnold W. M. Smeulders, Philip H. S. Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. In *ECCV (3)*, volume 11207 of *Lecture Notes in Computer Science*, pages 692–707. Springer, 2018. 1, 2, 3, 7, 8

[32] Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: visual tracking by re-detection. In *CVPR*, pages 6577–6587. IEEE, 2020. 2, 3, 8

[33] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic, faster and stronger. *CoRR*, abs/2003.10152, 2020. 3

[34] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *CoRR*, abs/1909.12605, 2019. 1, 2, 3, 6

[35] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418. IEEE Computer Society, 2013. 3, 8

[36] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *CVPR*, pages 3376–3385. IEEE Computer Society, 2017. 5

[37] Bin Yan, Haojie Zhao, Dong Wang, Huchuan Lu, and Xiaoyun Yang. 'skimming-perusal' tracking: A framework for real-time and robust long-term tracking. In *ICCV*, pages 2385–2393. IEEE, 2019. 2, 7, 8

[38] Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *NeurIPS*, pages 1305–1316, 2019. 2, 3

[39] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, pages 2403–2412. IEEE Computer Society, 2018. 4, 6

[40] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. A simple baseline for multi-object tracking. *CoRR*, abs/2004.01888, 2020. 2, 6

[41] Yunhua Zhang, Dong Wang, Lijun Wang, Jinqing Qi, and Huchuan Lu. Learning regression and verification networks for long-term visual tracking. *CoRR*, abs/1809.04320, 2018. 2, 8

[42] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, pages 4591–4600. Computer Vision Foundation / IEEE, 2019. 2

[43] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXI*, volume 12366 of *Lecture Notes in Computer Science*, pages 771–787. Springer, 2020. 2

[44] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. In *CVPR*, pages 3346–3355. IEEE Computer Society, 2017. 5

[45] Bineng Zhong, Bing Bai, Jun Li, Yulun Zhang, and Yun Fu. Hierarchical tracking by reinforcement learning-based searching and coarse-to-fine verifying. *IEEE Trans. Image Process.*, 28(5):2331–2341, 2019. 2

[46] Qinqin Zhou, Bineng Zhong, Xiangyuan Lan, Gan Sun, Yulun Zhang, Baochang Zhang, and Rongrong Ji. Fine-grained spatial alignment model for person re-identification with focal triplet loss. *IEEE Trans. Image Process.*, 29:7578–7589, 2020. 2

[47] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV (9)*, volume 11213 of *Lecture Notes in Computer Science*, pages 103–119. Springer, 2018. 2