Learning Localization-Aware Target Confidence for Siamese Visual Tracking

Jiahao Nie\textsuperscript{○}, \textit{Graduate Student Member, IEEE}, Zhiwei He\textsuperscript{○}, \textit{Member, IEEE}, Yuxiang Yang\textsuperscript{○}, \textit{Member, IEEE}, Mingyu Gao\textsuperscript{○}, and Zhekang Dong\textsuperscript{○}, \textit{Member, IEEE}

Abstract—Siamese tracking paradigm has achieved great success, providing effective appearance discrimination and size estimation by classification and regression. While such a paradigm typically optimizes the classification and regression independently, leading to task misalignment (accurate prediction boxes have no high target confidence scores). In this paper, to alleviate this misalignment, we propose a novel tracking paradigm, called SiamLA. Within this paradigm, a series of simple, yet effective localization-aware components are introduced to generate localization-aware target confidence scores. Specifically, with the proposed localization-aware dynamic label (LADL) loss and localization-aware label smoothing (LALS) strategy, collaborative optimization between the classification and regression is achieved, enabling classification scores to be aware of location state, not just appearance similarity. Besides, we propose a separate localization-aware quality prediction (LAQP) branch to produce location quality scores to further modify the classification scores. To guide a more reliable modification, a novel localization-aware feature aggregation (LAFQ) module is designed and embedded into this branch. Consequently, the resulting target confidence scores are more discriminative for the location state, allowing accurate prediction boxes to be predicted as high scores. Extensive experiments are conducted on six challenging benchmarks, including GOT-10k, TrackingNet, LaSOT, TNL2K, OTB100 and VOT2018. Our SiamLA achieves competitive performance in terms of both accuracy and efficiency. Furthermore, a stability analysis reveals that our tracking paradigm is relatively stable, implying that the paradigm is potential for real-world applications.

Index Terms—Localization-aware components, Siamese tracking paradigm, task misalignment.

I. INTRODUCTION

VISUAL object tracking (VOT) is a fundamental task in computer vision, and has been widely applied to human computer interfaces, autonomous driving and smart surveillance [1], [2]. Given an arbitrary target object in the initial frame, trackers need to tell us “where” this object will appear in subsequent frames [3]. Although great progress achieved, VOT is still a challenging task to tackle when faced with complex scenes, including occlusion, deformation, etc [4], [5], [6], [7], [8].

Recently, the Siamese tracking paradigm [9], [10], [11], [12], [13], [14] has attracted great attention from researchers, due to its well-balanced accuracy and efficiency. The Siamese architecture consists of two streams: template and search. The template stream extracts features of the given target object in the initial frame, and the search stream extracts features of subsequence frames. After feature extraction, a classification or regression structure follows. Early Siamese paradigm [9] contains only a classification branch, and the target size estimation is performed using a traditional multi-scale strategy. As deep learning techniques mature [15], [16], the modern Siamese paradigm can be classified into two categories: anchor-based [10] and anchor-free [11] paradigms. Both types contain classification and regression branches, which jointly perform the tracking task. The classification branch discriminates the appearance similarity based on the anchor or point features, and then outputs the target confidence scores. Meanwhile, the predicted box with the highest target confidence score, is obtained from the regression branch.

Although the development, however, such modern Siamese paradigms generally optimize the classification and regression independently in training, failing to make them synchronized, and thus causing a task-misalignment problem. The existing anchor-based paradigm [10], [17], [18] (Fig. 1(a)) has rarely investigated this misalignment, so the tracking performance often falls into a suboptimal state. As illustrated in Fig. 2(a), due to the task misalignment, the predicted box with high intersection over union (IoU) may not have high classification score, and the contribution of the classification is greater, because the optimal box given by the regression usually has a high IoU. As a more elegant structure, the anchor-free paradigm [11], [13] (Fig. 1(b)) attempts to address the task misalignment via a localization branch (i.e., center-ness). Intuitively, center-ness predicts the potential target centers, not the location accuracy. In addition, the output location quality scores are derived from the classification branch, which lack localization-aware information, resulting in the distractor being mistaken for the target object, as shown in Fig. 2(b). Therefore, the task misalignment problem is still not well resolved.

1520-9210 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Inspired by the above analysis, in this paper, we aim to alleviate the task misalignment, and mine the optimal prediction box as tracking box. To achieve this goal, we propose a novel Siamese tracking paradigm based on the anchor-free paradigm, as illustrated in Fig. 1(c). The key innovation lies in the prediction head, and a series of localization-aware components are developed to generate localization-aware target confidence scores. (i) Since the predicted box with the highest IoU may not have the highest classification score, we propose a localization-aware dynamic label (LADL) loss function to alleviate this situation. Benefit from this function, the classification and regression can be optimized collaboratively, leading to accurate prediction boxes being predicted as high scores. In detail, the location state information (i.e., IoUs) from the regression branch, is used to assign labels for classification samples, allowing the classification network to discriminate positive samples in a localization-aware manner, while not treating them equally. To facilitate the collaborative optimization, we further elaborate a localization-aware label smoothing (LALS) strategy to define the initial labels in the LADL loss function, which increases the probability of the optimal box being involved. This is quite effective and is not presented in the existing anchor-free trackers [11], [12], [13], [14]. (ii) Using the anchor-free tracking paradigm as the baseline, we exclusively design a separate localization-aware quality prediction (LAQP) branch to replace the center-ness to further alleviate the task misalignment. Our idea is to link this branch with the regression branch to model the localization-aware information, and then deliver the location quality scores. In practice, similar to classification training, by using IoUs from the regression branch as labels, we integrate the localization-aware information into the separate localization branch. Thanks to the separate structure, the learned features are dedicated, which fully embed localization-aware information into the localization network. When testing, the location quality scores and classification scores are combined to determine the target confidence scores, thus implicitly collaborating the classification and regression, and guiding more accurate target confidence scores for the predicted boxes. To improve the accuracy of the location quality scores, a localization-aware feature aggregation (LAFA) module is created to integrate more localization-aware prior into the LAQP branch. Within the LAFA, we develop a feature aggregation block to gather point-set features to represent the predicted boxes, rather than a single point feature, which lacks significant location state information. Moreover, considering that the boundary points have poor representation for the target object, we propose a localization-aware non-local block to capture the long- and short-term visual dependencies associated with the target object, enhancing the feature representation of the boundary points.

The main contributions of this work can be summarized as follows:

- We propose a novel Siamese tracking paradigm, called SiamLA. The paradigm alleviates the task misalignment in a simple yet effective, efficient and stable way.
- Following this paradigm, a series of localization-aware components are proposed, collaborating the classification and regression in the training and inference process.
- Our SiamLA outperforms the prevalent anchor-based and anchor-free paradigms, achieving competitive performance on several challenging benchmarks, while running at a real-time speed of 29 fps.

II. RELATED WORK

A. Siamese Network Based Framework in Visual Tracking

SiamFC [9] is a seminal Siamese tracking algorithm proposed by Bertinetto et al. Due to its well-balanced performance and efficiency, many excellent works [19], [20], [21], [22], [23],
[24] have emerged. E.g., [19] proposes a dual-margin Siamese network to learn general rules of target appearance variations. CFNet [20] integrates a correlation filter into a Siamese network structure, guiding a discriminative feature representation for target objects. SA-Siam [21] builds a twofold Siamese network to construct rich appearance and semantic information. StructSiam [22] extracts local structure information to identify discriminative regions of target objects. SiamDW [23] proposes a backbone that is more appropriate for tracking. Despite achieving promising results, these methods are rarely devoted to target size estimation, using only traditional multi-scale strategies. As deep learning techniques developed and given the relevance of object detection and object tracking tasks, a regional proposal network (RPN) [25] is introduced to estimate the target size in tracking, which makes the Siamese paradigm shine again. Unlike SiamFC-series trackers, SiamRPN [10] decomposes the tracking task into independent classification and regression. Benefit from the ability of the regression branch to model target shape changes, SiamRPN improves tracking accuracy, while remaining highly efficient. Based on this strong baseline, DaSiamRPN [17] enhances the anti-interference capability by sampling more hard negative samples. SiamRPN++ [18] replaces AlexNet [26] with ResNet [27] and uses a layer-wise aggregation strategy, also improving tracking performance. In addition, many methods [28], [29], [30], [31], [32], [33], [34], [35], [36] have investigated the feature fusion part, i.e., the neck of a model. After this anchor-based tracking period, the anchor-free tracking paradigm has become prevalent. Inspired by the anchor-free mechanism in detection, many anchor-free trackers [11], [12], [13], [14], [37], [38] have been designed. Compared with the anchor-based structure, anchor-free design discards the tracker-sensitive anchor hyper-parameter, leading to a more elegant and efficient tracking paradigm. In addition, an additional localization branch is employed in the anchor-free paradigm to predict the location quality of the predicted boxes. In this paper, the proposed Siamese tracking paradigm is based on the anchor-free paradigm, and aims to alleviate the task misalignment mentioned in Section 1.

B. Task Misalignment in Visual Tracking

Generally, modern Siamese trackers rely on the target confidence scores as a criterion for tracking. However, most of them optimize the classification and regression independently in training, inevitably resulting in predicted boxes with high target confidence scores may not be accurate. Towards this issue, some multi-stage methods [39], [40], [41] have been proposed. SPM [39] employs SiamRPN as the first stage to generate high quality proposals, followed by a lightweight network as the second stage to select the optimal one. C-RPN [40] iterates continuously to produce more accurate prediction boxes by a cascaded RPN. These multi-stage paradigm trackers are trained stage by stage and operate inefficiently. As a better solution, SiamRCR [42] implements task alignment by reweight the losses for each positive sample, allowing the model to pay more attention to high-quality samples with high IoUs. In contrast, we propose a localization-aware dynamic label (LADL) loss, making the classification and regression branches to be optimized collaboratively. In particular, we elaborate a simple, yet effective sampling strategy, localization-aware label smoothing (LALS). This promotes collaborative optimization, further increasing the probability of the optimal box being involved. To better address the task misalignment, we creatively propose a separate localization-aware quality prediction (LAQP) branch to modify the classification scores, implicitly collaborating the classification and regression. The work related to this branch is presented below.

C. Localization Branch in Visual Tracking

Localization branch serves as a unique structure in the anchor-free paradigm. SiamFC++ [13] and SiamCAR [11] use center-ness [43] to assess the location quality, and deliver the location quality scores to modify the classification scores. Nevertheless, the center-ness fails to address the task misalignment problem due to the lack of localization-aware information. Additionally, in the post-processing process, the penalty window plays a similar role to the center-ness, which downplays the role of the center-ness. To leverage the localization branch to alleviate the task misalignment, we abandon the center-ness design philosophy, and separate this branch. Besides, a localization-aware feature aggregation (LAFA) module is proposed within the LAQP branch.

III. PROPOSED METHOD

In this work, we propose a novel Siamese tracking paradigm, called SiamLA, as shown in Fig. 3. The implementation procedure is presented in this section. We briefly revisit the baseline tracker with the testing and training phases in section 3.1. Then, we detail the proposed LADL, LALS and LAFA in section 3.2, 3.3 and 3.4, respectively.

A. Revisit of the Baseline Method

Here, we employ a classic anchor-free tracker SiamCAR [11] as our baseline. The testing and training process are as follows. More details of testing and training can be found in [11].

Testing: As illustrated in Fig. 3, the template image \( z \) and search image \( x \) are fed into the Siamese backbone to extract the features \( \phi(z) \) and \( \phi(x) \). Then, a cross-correlation operator is applied between the features \( \phi(z) \) and \( \phi(x) \) to fuse them. This transfers the template information to the search region for subsequent tasks, classification and regression. The classification scores map \( c_{i,j} \), regression offset map \( r_{i,j} \) and location quality scores map \( C_{i,j} \) are calculated by

\[
\{c_{i,j}\} = f_{cls}(Cr_{cls}(\phi(z) \ast \phi(x))),
\{r_{i,j}\} = f_{reg}(Cr_{reg}(\phi(z) \ast \phi(x))),
\{C_{i,j}\} = f_{cen}(Cr_{cls}(\phi(z) \ast \phi(x)))
\]

(1)

where \( \ast \) denotes the cross-correlation operator, \( \phi \) represents the Siamese backbone, \( Cr \) is a series of “Conv-Relu-Bn’s”, and the convolution layer \( f_{cls/cen/reg}() \) outputs the corresponding map. Finally, the predicted box corresponding to the highest
Localization-Aware Target Confidence for Siamese Visual Tracking

NIE et al.: LEARNING LOCALIZATION-AWARE TARGET CONFIDENCE FOR SIAMESE VISUAL TRACKING 6197

Fig. 3. An overview of our Siamese tracking paradigm SiamLA, consisting of a Siamese backbone, feature fusion module and localization-aware prediction head. Thanks to the proposed localization-aware dynamic label (LADL) and localization-aware label smoothing (LALS), classification and regression can be optimized collaboratively. In addition, a separate localization branch, with a localization-aware feature aggregation (LAFA) module as its core is designed to implicitly collaborate the classification and regression.

target confidence score (i.e., the classification score multiplied by the location quality score) is considered as the tracking result.

**Training**: SiamCAR is trained with three losses: cross-entropy loss $L_{ce}$, IoU loss $L_{iou}$ and binary cross-entropy loss $L_{bce}$. They are used to optimize the classification, regression and center-ness branches, respectively, as follows

$$L_{cls} = \frac{1}{N_{cls}} \sum_{i,j} L_{ce}(c_{i,j}, \hat{c}_{i,j}),$$

$$L_{reg} = \frac{1}{N_{reg}} \sum_{i,j} \sum_{\hat{c}_{i,j}=1} L_{iou}(r_{i,j}, \hat{r}_{i,j}),$$

$$L_{cen} = \frac{1}{N_{cen}} \sum_{i,j} \sum_{\hat{c}_{i,j}=1} L_{bce}(C_{i,j}, \hat{C}_{i,j})$$

where $\hat{c}_{i,j}$, $\hat{r}_{i,j}$ and $\hat{C}_{i,j}$ represent the corresponding labels. For the classification, pixels within a certain size rectangle, centered on the target object are defined as positive samples, and the other pixels of the image are defined as negative samples. For the regression and center-ness, only positive samples are sampled for training.

**B. Localization-Aware Dynamic Label and Label Smoothing**

In the existing anchor-free trackers, the classification and regression independently optimize their respective loss objectives in training, resulting in task misalignment, i.e., the prediction boxes corresponding to points with high target confidence scores may not be accurate. However, we expect that the higher the target confidence score, the more accurate the corresponding prediction box. To satisfy this, the parallelized training style is discarded. We propose to use the output information from the regression branch as a training constraint term for the classification branch, thus achieving collaborative optimization and making the classification scores positively correlated with their corresponding IoUs. In practice, the IoUs of predicted boxes in the regression branch are adopted as dynamic labels for the classification samples. Here, we redefine the training objective for classification. This novel localization-aware dynamic label loss can be formulated as

$$\mathcal{L}_{ladl} = \frac{1}{N_{ladl}} \sum_{i,j} \text{LADL}(c_{i,j}, \hat{c}_{i,j})$$

where the LADL loss function as follow

$$\text{LADL}(c_{i,j}, \hat{c}_{i,j}) = \begin{cases} (1 - IoU_{i,j}) \log(1 - c_{i,j}) & \text{if } \hat{c}_{i,j} = 1, \\ + (1 - IoU_{i,j}) \log(1 - c_{i,j}) & \text{if } \hat{c}_{i,j} = 0 \end{cases}$$

where the predicted classification scores map $c \in \mathbb{R}^{H \times W \times 1}$, and the $IoU_{i,j}$ is calculated by

$$IoU_{i,j} = \frac{1}{1 + e^{(IoU_{i,j} - \alpha) \times \beta}}$$

where $IoU_{i,j} \in [0,1]$ and the parameters $\alpha$ and $\beta$ are set to 0.5 and 10 to guide the output $\hat{c}_{i,j} \in (0,1)$. (5) reduces the output variation at both ends, making the classification easier to optimize. Moreover, the low output variation at the end of high IoU, also increases the stability because the ground truth is annotated with reasonable noise. During the training, considering that no accurate prediction box is available in the initial stage, we normalize the dynamic labels at image-level by

$$\hat{IoU}_{i,j} = \frac{IoU_{i,j} - \min\{\{IoU_{i,j}\}\}}{\max\{\{IoU_{i,j}\}\} - \min\{\{IoU_{i,j}\}\}}$$

where the output is $\hat{IoU}_{i,j} \in [0,1]$. As illustrated in the proposed LADL (4), the loss for different samples is dynamically adjusted according to $\hat{IoU}_{i,j}$. The larger the $\hat{IoU}_{i,j}$, the higher
localization-aware label smoothing

On the left, initial labels for the samples are defined. On the right, we smooth the initial labels using IoUs and a hyper-parameter $\lambda$. $\lambda = 1$ represents the red points (positive samples), $\lambda = 0$ represents the blue points (negative samples). In terms of the green points (regularized positive samples), the smoothing curves at different $\lambda$ are presented.

Fig. 4. Illustration of the localization-aware label smoothing strategy. On the left, initial labels for the samples are defined. On the right, we smooth the initial labels using IoUs and a hyper-parameter $\lambda$. $\lambda = 1$ represents the red points (positive samples), $\lambda = 0$ represents the blue points (negative samples). In terms of the green points (regularized positive samples), the smoothing curves at different $\lambda$ are presented.

Although the above localization-aware dynamic label effectively alleviates the task misalignment, it still fails to mine the optimal prediction box, which may be regressed by the boundary points. This is because the boundary points are sampled as negative samples to train the classification network. To solve this problem, we elaborate a localization-aware label smoothing strategy to define the labels of the boundary points. As illustrated in Fig. 4(a), we consider the boundary (green points) as positive samples as well, and smooth their labels as $\lambda$. In fact, the target confidence scores of the center points are expected to be higher than those of the boundary points, which reduces the occurrence of tracker drift. Therefore, we distinguish between positive samples at the center and boundary points. In detail, for the center samples (red points), the labels are completely determined by the corresponding IoUs, while for the boundary samples (green points), a positive sample is considered only if the IoU is large enough, and its label should be smaller than the center sample with the same IoU. To implement this, we reformulate (5) as

$$IoU_{i,j} = \frac{1}{1 + e^{(1 - IoU_{i,j}) - \alpha \times \beta}}$$

(7)

where $\lambda = 1$ for center samples, other $\lambda$ for boundary samples. Fig. 4(b) shows the label changes under different $\lambda$, and it is evident that (7) satisfies our requirements. The parameter $\lambda$ is explicitly discussed in Section IV.

C. Localization-Aware Quality Prediction and Feature Aggregation

Most of the existing anchor-free trackers use center-ness as localization branch, which is induced from the classification branch. However, such a structure lacks localization-aware information, leaving the task-misalignment problem unresolved. Therefore, we design a separate localization-aware quality prediction branch, and supervise its training with localization-aware information from the regression branch. Besides, we propose a localization-aware feature aggregation module that contains a localization-aware non-local block and a feature aggregation block, guiding more accurate location quality scores.

In practice, as shown in Fig. 5, we propose a feature aggregation block to aggregate the features of the center, top-left, top-right, bottom-right, and bottom-left points of the predicted boxes, instead of relying on only the single center point features. Nevertheless, one shortcoming that exists in this block is that the boundary point feature has a weak representation for target objects. Thus, we develop an efficient localization-aware non-local block to capture the long- and short-term visual dependencies associated with the target objects. In contrast to the normal non-local structure, we incorporate a learnable position embedding, making this block aware of the location state. This is inspired by TrDiMP [44] and ToMP [45], we introduce positional prior knowledge to generate the position embedding. But differently, we employ a naïve correlation operator $h(\cdot)$ between the template and search features to highlight the target position, while not using a labeled mask presented in [44], [45]. A $1 \times 1$ convolution layer $e(\cdot)$ then is used to expand the channels as follows

$$p = e(h(\phi(x), \phi(x)))$$

(8)

The data flow of the whole localization-aware non-local block is depicted in Fig. 6, which can be expressed using an equation as

$$y = f(Softmax(\theta(x + p)\phi^\top(x + p)), \varphi(x)) + x$$

(9)

where $f(\cdot), \theta(\cdot), \varphi(\cdot)$ denote a $1 \times 1$ convolution layer, and $x$ and $p$ represent the input features and the introduced position embedding, respectively. To train the separate localization branch, similar to (4), we use IoUs as supervised information. Moreover, we sample the negative samples in a 1:1 (positive: negative) ratio to deepen the anti-distractors capability of the localization network. The optimization objective is formulated as

$$L_{laqp} = \frac{-1}{N_{laqp}} \sum_{i,j} L_{bec}(c_{i,j}, IoU_{i,j})$$

(10)

For testing, the location quality scores serve as an effective constraint on the classification scores to guide the more accurate target confidence scores. This implicitly collaborates the classification and regression, thus also alleviating the task misalignment to some extent.
the tracking process is treated as a point selection by the target confidence scores. The primary target confidence scores are determined by a 2D vector \((cls_{i,j}, loc_{i,j})\), in which \(cls_{i,j}\) and \(loc_{i,j}\) represent the outputs of the classification and localization branches, respectively. Based on the fact that subtle target changes occur in adjacent frames, we use two penalty factors to rerank the target confidence scores, as follows

\[
s_{i,j} = (1 - w)cls_{i,j} \times loc_{i,j} \times p_{i,j} + wW_{i,j} \quad (12)
\]

where \(W\) is the cosine window, weighted by a window influence weight \(w\); \(p_{i,j}\) is the penalty factor for scale change, which is formulated by

\[
p_{i,j} = e^{-k \times \max\left(\frac{r}{b}, \frac{r'}{b'}\right) \times \max\left(\frac{s}{s'}, \frac{s}{s'}\right)} \quad (13)
\]

where \(k\) is a hyper-parameter, \(r\) and \(r'\) represent the aspect ratio \(\frac{s}{s'}\) of the predicted boxes in the current and previous frames, \(b\) and \(b'\) represent the sizes of the predicted boxes. The mean of the predicted boxes corresponding to the top-n target confidence scores \(s_{i,j}\) is regarded as the tracking result, and \(n\) is set to 3 as in [11]. The entire tracking process is presented in Algorithm 1.

**Evaluation benchmarks and metrics:** We evaluate the proposed tracker SiamLA on six challenging benchmarks: GOT-10k [48], TrackingNet [50], LaSOT [51], TNL2K [52], OTB100 [53] and VOT2018 [54]. GOT-10k uses the average overlap (AO) and success rate (SR) to evaluate the performance. For the TrackingNet and LaSOT benchmarks, three metrics, i.e., area under the curve (AUC), precision (Prec.), plus the normalized precision (N_Prec.) are adopted. Similarly, on the TNLK2 and OTB100 benchmarks, the trackers are ranked using AUC and Prec. scores. All of these benchmarks use a one-pass-evaluation (OPE) to run a tracker. For the VOT2018 benchmark, whenever tracking fails, a tracker will reinitialize a new frame as the template image. Based on this running style, two metrics, accuracy and robustness are derived to evaluate the performance. In addition, the expected average overlap (EAO) is a comprehensive metric that takes both accuracy and robustness into account.

**Algorithm 1:** Tracking with SiamLA.

**Input:** Images of a video sequence \(\{I_t\}_{t=1}^T\): The given target box \(b_1\) in initial frame \(I_1\).

**Output:** Predicted boxes \(\{b_t\}_{t=2}^T\) in subsequences frames.

1. Crop the template image \(I_t\) using \(b_1\) for \(I_1\).
2. Extract feature \(\phi(z)\) of \(z\).
3. for \(t = 2\) to \(T\) do
4. Crop the search image \(x\) using \(b_{t-1}\) for \(I_t\);
5. Extract feature \(\phi(x)\) of \(x\);
6. Obtain the target confidence scores \(\{s_{i,j}\}\) by (13);
7. Obtain \(b_t\) using \(\{s_{i,j}\}\) and the post-processing algorithm in SiamCAR [11];
8. end for

**IV. EXPERIMENTS**

**A. Implementation Details**

**Training details:** The modified ResNet-50 [27], that is initialized with the parameters pre-trained on ImageNet [46], is employed as the backbone network of the proposed tracker SiamLA. Following the setting of the anchor-free trackers [11], [12], [13], [14], we use the training splits of the Youtube-BB [47], ImageNet VID [46], ImageNet DET [46], GOT-10k [48] and COCO [49] datasets to train the model offline. The template and search images are cropped to 127 pixels and 255 pixels centered on the target objects. The overall training objective can be defined as

\[
L = L_{ladl} + \lambda_1 L_{reg} + \lambda_2 L_{laqp} \quad (11)
\]

where \(\lambda_1\) and \(\lambda_2\) are weights, both set to 1 to balance the different training losses. An SGD optimizer is utilized to optimize this overall training objective, and the batch size is set to 128 images.

With a total of 50 epochs trained, for the first 5 epochs, we freeze the backbone and train the localization-aware prediction head. For the remaining epochs, the whole model is trained end-to-end with an exponential decay of the learning rate from \(10^{-3}\) to \(10^{-5}\). Our code is implemented using PyTorch 1.8.0 and Python 3.6. The experiments are conducted on a workstation with an Intel i7-10700F @ 2.9 GHz CPU, and 4 NVIDIA GTX 1080Ti GPUs with CUDA10.2.

**Testing details:** For a fair comparison, we use the same testing strategy as SiamCAR [11], to evaluate the proposed tracker SiamLA. Since the anchor-free tracking paradigm is inherited, the tracking algorithm in SiamCAR [11]; the entire tracking process is presented in Algorithm 1.
B. Comparison With State-of-the-art Methods

**GOT-10 k:** GOT-10k [48] is a large-scale generic object tracking benchmark, and the test set contains 180 video sequences with 84 object classes. The ground truths are not open, so the performance of a tracker needs to be obtained by submitting the results to an evaluation service platform. We follow the protocols in [48], and use only the GOT-10 k training set to train our SiamLA. As shown in Table I, the proposed tracker SiamLA achieves favorable performance, obtaining 0.619, 0.724 and 0.510 scores in terms of AO, SR\_0.5, and SR\_0.75, respectively. Although SiamRCR and SiamGAT exhibit a slightly inferior performance in AO or SR\_0.5, our SiamLA has a significant advantage in SR\_0.75. Compared with the baseline SiamCAR, our SiamLA improves AO and SR\_0.5 scores by 3.8 and 4.1 points. Especially for SR\_0.75, with the IoU threshold set to 0.75, our tracking success rate improves by 6.9 points, indicating that the task misalignment is effectively alleviated, i.e., the proposed Siamese tracking paradigm is able to mine the optimal prediction boxes.

**TrackingNet:** TrackingNet [50] is a large-scale short-term tracking benchmark that is collected from the real world, and has a test set of 511 video sequences. Similar to GOT-10 k, the tracking results need to be evaluated on a specific service platform. The performance comparison on this benchmark is presented in Table I. Our tracker SiamLA achieves the best performance among all compared state-of-the-art trackers, obtaining 0.767, 0.718 and 0.821 scores in terms of AUC, Prec. and N_\_Prec., respectively. Although the baseline SiamCAR exhibits impressive tracking performance, our SiamLA still outperforms it by 2.7, 3.4 and 1.7 points in the three metrics. Since the various wild scenes are involved in the test set, the excellent performance demonstrates the effectiveness and generality of the proposed tracking paradigm in the real world.

**LaSOT:** To further evaluate the proposed tracking paradigm, we conduct experiments on the challenging long-term benchmark LaSOT [51], which consists of 1400 video sequences (280 for the test set) with 3.52 million frames. It is labeled with 14 different attributes. We report the results in Table I. The proposed tracker SiamLA achieves the competitive performance. Compared with the classic anchor-free trackers SiamCAR [11], Ocean [12], SiamFC++ [13] and SiamBAN [14], our SiamLA performs better in terms of both accuracy and efficiency, running at a real-time speed of 29 fps. Notably, we obtain the best N_\_Prec. score of 0.652. Moreover, we select some state-of-the-art trackers to report a comparative analysis of 14 different attributes. As shown in Fig. 7, the proposed SiamLA achieves optimal performance for 10 attributes. Compared with the baseline SiamCAR, we obtain performance improvements in 13 attributes ranging from 3% (rotation) to 8% (camera motion). Thanks to the proposed separate localization branch,

### Table I

**State-of-the-Art Comparison on GOT-10k [48], TrackingNet [50] and LaSOT [51]. The Best Three Results Are Colored by Red, Green and Blue**

| Paradigm      | Tracker     | GOT-10k [48] | TrackingNet [50] | LaSOT [51] | Speed |
|---------------|-------------|--------------|-------------------|------------|------|
|               |             | AO | SR\_0.5 | SR\_0.75 | AUC | Prec. | N_\_Prec. | AUC | Prec. | N_\_Prec. | Device | Fps  |
| SiamFC-series | SiamFC [9]  | 0.392 | 0.426 | 0.135 | 0.571 | 0.533 | 0.654 | 0.336 | 0.339 | 0.420 | Titan X 86 |
|               | CFNet [20]  | 0.434 | 0.481 | 0.190 | 0.578 | 0.533 | 0.654 | 0.275 | 0.259 | 0.312 | Titan X 52 |
|               | DSiam [55]  | 0.417 | 0.461 | 0.149 | 0.333 | -     | 0.405 | 0.333 | 0.322 | 0.405 | Titan X 45 |
|               | StructSiam  | -   | -     | -     | -   | -     | -     | -    | -    | -     | -     |
|               | SiamDW [23] | 0.411 | 0.456 | 0.154 | -   | -     | -     | -    | -    | -     | -     |
| Anchor-based  | SiamRPN [10]| 0.444 | 0.536 | 0.222 | -   | -     | -     | 0.447 | 0.432 | 0.542 | GTX 1060 X |
|               | DaSiamRPN [17]| 0.481 | 0.581 | 0.270 | 0.638 | 0.591 | 0.733 | 0.515 | 0.539 | 0.605 | Titan X 110 |
|               | SiamRPN++ [18]| 0.518 | 0.618 | 0.325 | 0.733 | 0.694 | 0.800 | 0.496 | 0.491 | 0.569 | Titan Xp 35 |
|               | SiamLTR [56]| 0.593 | 0.698 | 0.474 | 0.736 | 0.691 | 0.802 | 0.525 | 0.533 | -     | GTX 1080Ti |
|               | SPM [39]    | 0.513 | 0.593 | 0.359 | 0.712 | 0.660 | 0.771 | -    | -    | -     | Tesla P100 120 |
|               | C-RPN [40]  | -   | -     | -     | 0.669 | 0.619 | 0.746 | 0.455 | 0.425 | -     | GTX 1080Ti 36 |
|               | SPLIT [57]  | -   | -     | -     | 0.426 | 0.396 | 0.494 | -    | -    | -     | GTX Titan X 26 |
|               | DS [58]     | 0.597 | 0.676 | 0.415 | 0.728 | 0.664 | 0.768 | 0.488 | 0.490 | -     | GTX 1080 25 |
|               | ROAM [59]   | 0.465 | 0.532 | 0.236 | 0.670 | 0.623 | 0.754 | 0.447 | 0.445 | -     | RTX 2080 13 |
| Anchor-free   | SiamBAN [14]| 0.579 | 0.684 | 0.457 | -   | -     | -     | 0.514 | 0.521 | 0.598 | GTX 1080Ti 40 |
|               | SiamCAR     | 0.581 | 0.683 | 0.441 | 0.740 | 0.684 | 0.804 | 0.516 | 0.524 | 0.610 | GTX 1080Ti 38 |
|               | SiamFC++ [13]| 0.595 | 0.695 | 0.497 | 0.754 | 0.705 | 0.800 | 0.544 | 0.547 | 0.623 | RTX 2080Ti 90 |
|               | Ocean [12]  | 0.611 | 0.721 | 0.473 | -   | -     | -     | 0.560 | 0.566 | 0.651 | Tesla V100 25 |
|               | SiamRCR [42]| 0.624 | 0.752 | 0.460 | 0.764 | 0.716 | 0.818 | 0.575 | 0.599 | -     | Tesla P40 65 |
|               | SiamGAT [30]| 0.627 | 0.743 | 0.488 | 0.753 | 0.698 | 0.807 | 0.539 | 0.530 | 0.633 | RTX 2080Ti 70 |
|               | SBT-light [60]| 0.602 | 0.685 | 0.530 | -   | -     | -     | 0.565 | 0.571 | -     | Tesla V100 62 |
| Ours          | SiamLA      | 0.619 | 0.724 | 0.510 | 0.767 | 0.718 | 0.821 | 0.561 | 0.560 | 0.652 | GTX 1080Ti |

---

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 8. Quantitative comparison of our SiamLA, SiamCAR [11], Ocean [12] and SiamBAN [14]. The proposed tracking paradigm, with localization-aware components, robustly predicts more accurate boxes.

Fig. 9. Precision and success plot on the TNL2K [52] test set (top) and OTB100 [53] (bottom), using one-pass evaluation. In the legend, the distance precision at the threshold of 20 pixels (Prec.) in the precision plot and area under the curve (AUC) in the success plot are marked.

Fig. 8: Quantitative comparison of our SiamLA, SiamCAR [11], Ocean [12] and SiamBAN [14]. The proposed tracking paradigm, with localization-aware components, robustly predicts more accurate boxes.

Fig. 9: Precision and success plot on the TNL2K [52] test set (top) and OTB100 [53] (bottom), using one-pass evaluation. In the legend, the distance precision at the threshold of 20 pixels (Prec.) in the precision plot and area under the curve (AUC) in the success plot are marked.

Table II

| Tracker    | EAO↑ | Accuracy↑ | Robustness↓ |
|------------|------|-----------|-------------|
| SiamFC [9] | 0.188| 0.503     | 0.585       |
| DaSiamRPN [10] | 0.384| 0.588     | 0.276       |
| Siam R-CNN [61] | 0.408| 0.609     | 0.220       |
| SiamRPN++ [18] | 0.414| 0.600     | 0.234       |
| SiamCAR [11] | 0.423| 0.574     | 0.197       |
| SiamFC++ [13] | 0.426| 0.587     | 0.183       |
| PGNet [28] | 0.447| 0.618     | 0.192       |
| STMTracker [62] | 0.447| 0.590     | 0.159       |
| SiamBAN [14] | 0.452| 0.597     | 0.178       |
| SiamRCR [42] | 0.457| 0.588     | 0.188       |
| TrDiMP [44] | 0.462| 0.600     | 0.141       |
| SiamLA | 0.462| 0.593     | 0.136       |

TNL2K: TNL2K [52] is a recently proposed large-scale benchmark for tracking by natural language and bounding box initialization, with more difficult video sequences. We use the bounding box initialization to evaluate our tracker on 700 video sequences. As shown in Fig. 9, the proposed SiamLA obtains the best AUC score of 0.419, and exceeds 6.3/3.9 points on AUC and Prec., compared to the baseline SiamCAR. In addition, we select the top 10 performance trackers for comparison of 14 attributes. As illustrated in Fig. 10, among the 14 attributes, except for low resolution, our SiamLA outperforms all compared trackers, proving that the proposed tracking paradigm is capable of handling various complex scenes.

OTB100: In addition to the above large-scale benchmarks, we evaluate the proposed tracker SiamLA on several popular small benchmarks. OTB100 [53] is one of the most common benchmarks used in the tracking community, containing 100 video sequences with 11 types of attributes. As shown in Fig. 9, the SiamLA achieves advanced performance of 0.709 and 0.929 scores on AUC and Prec. by the proposed localization-aware components.

VOT2018: VOT2018 [54] is also a small benchmark. It contains a total of 60 video sequences, serving as a benchmark for the seventh visual object tracking challenge. We conduct two sets of experiments to evaluate our tracker SiamLA. A comparison with state-of-the-art trackers is presented in Table II, the proposed SiamLA obtains scores of 0.462, 0.593 and 0.136 in terms of EAO, accuracy and robustness, respectively. Then, in Fig. 11, we exhibit an EAO ranking, and the proposed tracker SiamLA far outperforms the winner LADCF [63] in VOT2018.

C. Ablation Study

Component-wise analysis: To verify the effectiveness of the proposed localization-aware components (i.e., LADL, LALS, LAQP and LAFA), and investigate their contribution to the proposed tracker SiamLA, we conduct an ablation experiment on the OTB100 [53] and GOT-10k [48] benchmarks. Moreover, we also compare the running speed of the tracking variations with different components. To be fair, we train the tracking variations using the GOT-10k training set, and evaluate them with the same hyper-parameters in testing. We report the ablation results in Table III. The proposed LADL enables collaborative optimization between the classification and regression, making the accurate prediction boxes predict high target confidence scores. Thus, 2.4% and 1.3% improvements in AUC and AO are obtained, respectively. Besides, no speed loss is involved, because the LADL component has no additional time-space complexity in testing. When the LALS is used to define the initial labels in the LADL loss function, the probability of the optimal box being predicted as the result increases, further alleviating the task-misalignment problem and thus the performance is enhanced. The proposed separate LAQP branch delivers the location quality scores to
modify the classification scores, also leading to significant performance improvements, in which the AUC and AO scores are 2.0% and 1.2%. This is due to that this branch effectively modify the classification scores, implicitly collaborating the classification and regression. When we use the LAFA module to model localization-aware information in the LAQP branch, the performance improvements are further increased to 2.9% and 1.8% in terms of AUC and AO, respectively. Moreover, despite slower 9 fps than the baseline, it still exceeds the real-time speed (24 fps). With all these proposed components, our tracker achieves the best performance of 0.649 and 0.619 in terms of AUC and AO scores, which confirms the effectiveness of the proposed localization-aware components in this work.

To intuitively present how the proposed tracking paradigm alleviates task misalignment, we perform some visualization comparison experiments. As stated in the testing details, the selection of the predicted box depends on the target confidence scores. Therefore, a positive correlation between the IoUs and the target confidence scores implies perfect task alignment. As shown in Fig. 12, our tracker has a higher positive correlation than the baseline, indicating that the proposed localization-aware components effectively alleviate the task misalignment. To further visualize the effect of the proposed LADL and LALS, we plot the IoUs of predicted boxes and optimal boxes on two video sequences. As shown in Fig. 13, with the proposed LADL and LALS, the IoU curves of the predicted boxes better fit the IoU curves of the optimal boxes, confirming the effectiveness of the proposed localization-aware components in this work.
curve of the optimal boxes, which demonstrates that the LADL can guide high target confident scores for accurate predicted boxes, and LALS increases the optimal boxes being predicted as tracking boxes. Besides, since our model focus more on the samples with large IoUs by using the LADL loss, the optimal boxes in the right sub-figures are higher than that of the left sub-figures. In addition, we visualize the location quality scores generated by the center-ness and the proposed separate branch. As illustrated in Fig. 14, the top line shows that the center-ness predicts the potential target center, not the location accuracy. In contrast, our method predicts more high score points, indicating the classification easier to optimize, and thus leading to a performance increase. To prove this, we conduct a comparison experiment of the linear IoU and nonlinear IoU. The results presented in Table IV demonstrate the effectiveness of the nonlinear design in (5).

**Localization-aware dynamic label:** As illustrated in (5), we map the linearly varying $I_{i,j}$ to the nonlinear $I_{i,j}$, making the classification easier to optimize, and thus leading to a performance increase. To prove this, we conduct a comparison experiment of the linear IoU and nonlinear IoU. The results presented in Table IV demonstrate the effectiveness of the nonlinear design in (5).

**Localization-aware label smoothing:** We also exhibit a comparison experiment of the parameter $\lambda$ in (7), a significant hyper-parameter in the proposed LALS strategy. A larger $\lambda$ implies that more high-quality samples will be included. However, more distractors are introduced at the same time, preventing the model from selecting the most accurate prediction box. As shown in Table V, when $\lambda = 0.2$, the tracker performs best, obtaining 0.632 and 0.602 scores in terms of AUC and AO scores. Therefore, in all experiments, we set $\lambda$ to 0.2 if not specified.

### Table IV

| Tracking Variation     | AUC  | AO  |
|------------------------|------|-----|
| Linear $I_{i,j}$       | 0.613| 0.591|
| Nonlinear $I_{i,j}$    | 0.621| 0.594|

### Table V

| $\lambda$ | AUC  | AO  |
|-----------|------|-----|
| 0.4       | 0.604| 0.583|
| 0.3       | 0.624| 0.594|
| 0.2       | 0.632| 0.602|
| 0.1       | 0.619| 0.591|

### Table VI

| Tracking Variation               | AUC  | AO  |
|----------------------------------|------|-----|
| LAFA w/o localization-aware non-local block | 0.626| 0.599|
| LAFA w/o feature aggregation block | 0.611| 0.591|

_D. Stability Analysis_

In fact, the Siamese paradigm tend to be not stable enough in various scenes because some hyper-parameters in the post-processing algorithm need to be adapted to different scenes, making it difficult to apply to real-world. The window influence weight is a crucial hyper-parameter, as shown in (12). We conduct an extra experiment to demonstrate the stability of the proposed tracking paradigm. Fig. 15 presents the comparison results. We clearly observe that our tracker SiamLA is more stable than the baseline SiamCAR. Especially without post-processing in testing (i.e., window influence = 0), our tracker still shows competitive performance (0.687 AUC score), suggesting that the proposed tracking paradigm is superior at mining the optimal prediction box, making it more potential to real-world applications.

### E. Limitation and Discussion

Due to the Siamese network based framework we inherit, it is hard for the proposed SiamLA paradigm to perform a favorable appearance matching in the occlusion scene. As shown in Fig. 16, when the target cat (top line) and bus (bottom line) are occluded by disturbing objects, the accuracy of tracking boxes...
tracking paradigm. We also conduct some visualization experiments to clearly present how the paradigm alleviates the task misalignment. Moreover, a stability analysis reveals that our tracking paradigm is relatively stable.

**REFERENCES**

[1] Z. Sun, J. Chen, L. Chao, W. Ruan, and M. Mukherjee, “A survey of multiple pedestrian tracking based on tracking-by-detection framework,” IEEE Trans. Circuits Syst. Video Techn., vol. 31, no. 5, pp. 1819–1833, May 2021.

[2] J. Zhang and D. Tao, “Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things,” IEEE Internet Things J., vol. 8, no. 10, pp. 7789–7817, May 2021.

[3] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, “Deep learning for visual tracking: A comprehensive survey,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 5, pp. 3943–3968, May 2022.

[4] M. Kristan et al., “The eighth visual object tracking VOT2020 challenge results,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 547–601.

[5] M. Kristan et al., “The ninth visual object tracking VOT2021 challenge results,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 2711–2738.

[6] Y. Wu, J. Lim, and M. Yang, “Online object tracking: A benchmark,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2013, pp. 2411–2418.

[7] M. Mueller, N. Smith, and B. Ghanem, “A benchmark and simulator for uav tracking,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 445–461.

[8] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, “Need for speed: A benchmark for higher frame rate object tracking,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 1134–1143.

[9] L. Bertinetto et al., “Fully-convolutional siamese networks for object tracking,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 850–865.

[10] 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., 2018, pp. 8971–8980.

[11] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, “SiamCAR: Siamese fully convolutional classification and regression for visual tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6268–6276.

[12] Z. Zhang et al., “Ocean: Object-aware anchor-free tracking,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 771–787.

[13] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, “SiamFC: Towards robust and accurate visual tracking with target estimation guidelines,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 12549–12556.

[14] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, “Siamese box adaptive network for visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6667–6676.

[15] Y. Xu et al., “ViTAE: Vision transformer advanced by exploring intrinsic inductive bias,” in Proc. Adv. Neural Inf. Process. Syst., 2021, vol. 34, pp. 28522–28535.

[16] Q. Zhang et al., “Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond,” 2022. [Online]. Available: https://arxiv.org/abs/arxiv:2202.10108

[17] Z. Zhu et al., “Distractor-aware siamese networks for visual object tracking,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 101–117.
[18] B. Li et al., “SiamRPN : Evolution of siamese tracking with very deep networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4277–4286.

[19] N. Fan, X. Li, Z. Zhou, Q. Liu, and Z. He, “Learning dual-margin model for visual tracking,” in Neural Netw., vol. 140, pp. 344–354, 2021.

[20] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, “End-to-end representation learning for correlation filter based tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2805–2813.

[21] A. He, C. Luo, X. Tian, and W. Zeng, “A twofold siamese network for real-time object tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4834–4843.

[22] Y. Zhang et al., “Structured siamese network for real-time visual tracking,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 351–366.

[23] Z. Zhang et al., “Deeper and wider siamese networks for real-time visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4591–4600.

[24] Y. Zha, T. Ku, Y. Li, and P. Zhang, “Deep position-sensitive tracking,” IEEE Trans. Multimedia, vol. 22, no. 1, pp. 96–107, Jan. 2020.

[25] R. Girshick et al., “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1440–1448.

[26] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, Art. no. 25.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[28] B. Liao et al., “PG-Net: Pixel to global matching network for visual tracking,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 429–444.

[29] W. Han, X. Dong, F. S. Khan, L. Shao, and J. Shen, “Learning to fuse asymmetric feature maps in siamese trackers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 16570–16580.

[30] D. Guo et al., “Graph attention tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 9538–9547.

[31] X. Chen et al., “Transformer tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 8122–8131.

[32] Z. Zhang, Y. Liu, X. Wang, B. Li, and W. Hu, “Learn to match: Automatic matching network design for visual tracking,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 13339–13348.

[33] J. Nie, H. Wu, Z. He, M. Gao, and Z. Dong, “Spreading fine-grained prior knowledge for accurate tracking,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 9, pp. 6186–6199, Sept. 2022, doi: 10.1109/TCYTV.2022.3162599.

[34] Q. Liu et al., “Learning deep multi-level similarity for thermal infrared object tracking,” IEEE Trans. Multimedia, vol. 23, pp. 2114–2126, 2021.

[35] Q. Liu et al., “Learning dual-level deep representation for thermal infrared tracking,” IEEE Trans. Multimedia, to be published, doi: 10.1109/TMM.2022.3140929.

[36] M. Li, L. Peng, T. Wu, and Z. Peng, “A bottom-up and top-down integration framework for online object tracking,” IEEE Trans. Multimedia, vol. 23, pp. 105–119, 2021.

[37] K. Yang et al., “SiamCorners: Siamese corner networks for visual tracking,” IEEE Trans. Multimedia, vol. 24, pp. 1956–1967, 2022.

[38] S. Zhang, X. Zhao, and L. Fang, “CAT: Corner aided tracking with deep regression network,” IEEE Trans. Multimedia, vol. 23, pp. 859–870, 2021.

[39] G. Wang, C. Luo, Z. Xiong, and W. Zeng, “SPM-Tracker: Series-parallel matching for real-time visual object tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3638–3647.

[40] H. Fan and H. Ling, “Siamese cascaded region proposal networks for real-time visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7944–7953.

[41] G. Han, J. Su, Y. Liu, Y. Zhao, and S. Kwong, “Multi-stage visual tracking with siamese anchor-free proposal network,” IEEE Trans. Multimedia, to be published, doi: 10.1109/TMM.2021.3127357.

[42] J. Peng et al., “SiamRCC: Reciprocal classification and regression for visual object tracking,” 2021. [Online]. Available: https://arxiv.org/abs/2105.11237.

[43] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9627–9636.

[44] N. Wang, W. Zhou, J. Wang, and H. Li, “Transformer meets tracker: Exploiting temporal context for robust visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 1571–1580.

[45] C. Mayer et al., “Transforming model prediction for tracking,” 2022. [Online]. Available: https://arxiv.org/abs/2203.11192

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Yuxiang Yang (Member, IEEE) received the B.S. and Ph.D. degrees in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2008 and 2013, respectively. He joined Hangzhou Dianzi University, Hangzhou, China, in 2013, where he is currently an Associate Professor with the School of Electronic and Information. He was also a Visiting Scholar with The University of Sydney, Camperdown NSW, Australia, in 2020. He has authored or coauthored more than 30 papers on journals such as IJCV, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, KBS, and conferences such as CVPR, IJCAI, ICME. His research interests include computer vision and deep learning.

Mingyu Gao received the M.S. degree in power electronics from Zhejiang University, Hangzhou, China, in 1993, and the Ph.D. degree information and communication engineering from the Wuhan University of Technology, Wuhan, China, in 2013, in. He joined Hangzhou Dianzi University, in 2001, where he is currently a Professor with the School of Electronic and Information. He is the Director of Zhejiang Provincial Key Lab of Equipment Electronics and also the Deputy Leader of Zhejiang New Energy Vehicle and Intelligent Transportation Major Special Expert Group. His research interests include the areas of deep learning and computer vision.

Zhekang Dong (Member, IEEE) received the B.E. and M.E. degrees in electronics and information engineering in 2012 and 2015, respectively, from Southwest University, Chongqing, China. He received the Ph.D. degree from the School of Electrical Engineering, Zhejiang University, China, in 2019. He is currently the Professor with Hangzhou Dianzi University, Hangzhou, China. He is also the Research Assistant (Joint-Supervision) with The Hong Kong Polytechnic University. He has authored or coauthored more than 20 papers in refereed journals and conferences. His research interests include memristor and memristive system, artificial neural network, the design and analysis of nonlinear systems based on memristor and computer simulation.