Computers, Materials & Continua
DOI: 10.32604/cmc.2023.036177
Article

Tech Science Press

SiamDLA: Dynamic Label Assignment for Siamese Visual Tracking

Yannan Cai, Ke Tan and Zhenzhong Wei*

Key Laboratory of Precision Opto-Mechatronics Technology, Ministry of Education, School of Instrumentation Science and Opto-Electronics Engineering, Beihang University, Beijing, 100191, China
*Corresponding Author: Zhenzhong Wei. Email: zhenzhongwei@buaa.edu.cn

Received: 20 September 2022; Accepted: 08 December 2022

Abstract: Label assignment refers to determining positive/negative labels for each sample to supervise the training process. Existing Siamese-based trackers primarily use fixed label assignment strategies according to human prior knowledge; thus, they can be sensitive to predefined hyperparameters and fail to fit the spatial and scale variations of samples. In this study, we first develop a novel dynamic label assignment (DLA) module to handle the diverse data distributions and adaptively distinguish the foreground from the background based on the statistical characteristics of the target in visual object tracking. The core of DLA module is a two-step selection mechanism. The first step selects candidate samples according to the Euclidean distance between training samples and ground truth, and the second step selects positive/negative samples based on the mean and standard deviation of candidate samples. The proposed approach is general-purpose and can be easily integrated into anchor-based and anchor-free trackers for optimal sample-label matching. According to extensive experimental findings, Siamese-based trackers with DLA modules can refine target locations and outperform baseline trackers on OTB100, VOT2019, UAV123 and LaSOT. Particularly, DLA-SiamRPN++ improves SiamRPN++ by 1% AUC and DLA-SiamCAR improves SiamCAR by 2.5% AUC on OTB100. Furthermore, hyper-parameters analysis experiments show that DLA module hardly increases spatio-temporal complexity, the proposed approach maintains the same speed as the original tracker without additional overhead.

Keywords: Siamese network; label assignment; single object tracking; anchor-based; anchor-free

1 Introduction

Generic object tracking, which automatically calculates the trajectory of an arbitrary target in a changing video sequence, is a long-standing task in real-world computer vision applications such as self-driving cars, aerial-view tracking, surveillance and human-computer interactions [1,2]. However, the highly sought-after applications are limited by unpredictable object motion and appearance variance.

Deep learning has achieved great success and been applied to many real-world tasks such as object detection [3], contextual analysis [4], disaster management [5], person Re-ID [6], and Vehicle Re-ID [7]. In recent years, siamese neural networks have been widely used for visual object tracking. End-to-end offline training on large-scale image pairs by developing Siamese-based tracking tasks as a target-matching problem is a key to success [8]. The template patch and search region are the two branches that make up most Siamese-based trackers. The two branches calculate the same function to produce a similarity map where the template patch encodes object appearance information. However, the similarity matching function only acquires the deep structural general relation between the two branches and cannot cope with target scale and aspect ratio variations [9]. Thus, there are several potential solutions proposed to address this issue and the development of the Siamese-based tracking community is often accompanied by the enhancement of target state estimation fashions. Generally, target state estimation is reduced to regressing the bounding box, which best fits the target in the current frame. The existing methods may be summarized as multi-scale search, anchor-based, and anchor-free regression. Early studies [10–15] represented by SiamFC [10] rely on the coarse target localization provided by the similarity map, then use a brutal multi-scale test to estimate the object size, which is inaccurate and time-consuming. SiamRPN tracker family [16–18] introduces region proposal network (RPN) [19], originally employed for object detection, into the object tracking community. More concretely, they tile a group of predefined anchor boxes per location on the similarity map, and simultaneously perform foreground-background classification and object bounding box refinement, achieving a previously unseen performance. They have a deep influence on the next series of anchor-based siamese trackers [20–23]. Later anchor-free siamese trackers [24–28] emerge in a multitude tackling the object bounding box estimation in a per-pixel-prediction way, referring to FCOS [3] in object detection, avoiding heuristic hyper-parameter tuning about the anchors.

Researchers tend to intuitively believe that the performance gap between anchor-based and anchor-free trackers is obtained from different target state estimation fashions. However, a previous study [29] shows that the essential difference between center-based anchor-free and one-stage anchor-based detectors comes from the definition of positive/negative training samples. Greatly inspired by this, we studied the impact of label assignment on the performance of the tracker and investigated how to further enhance the performance without introducing additional overhead by reasonably defining positive/negative training samples. In this study, to avoid loss generality, we use the classical anchor-based tracker SiamRPN++ [18] and anchor-free tracker SiamCAR [26] as baselines. The two representative trackers are simple in structure and outstanding in performance. Locating the center point of the anchor box or grid cell corresponding to the search region is firstly needed to regress the object-bounding box for both SiamRPN++ [18] and SiamCAR [26]. This follows that accurate object state regression has a strong dependence on robust location classification. There are numerous ways to enhance classification accuracy, but few studies w.r.t refining label assignment. Meanwhile, the performance of SiamRPN++ [18] is very sensitive to the hyperparameter IoU threshold in our experiments. Here, IoU is the area ratio of intersection over the union of the anchor box and corresponding ground truth bounding box (*gt*). They define positive samples as $IoU > 0.6$ while negative samples as $IoU < 0.3$. As stated in Cascade R-CNN [30] and Ocean [27], trackers trained with a high IoU threshold vanish many positive samples, especially making it difficult to refine anchor boxes whose overlap with the *gt* is small. However, trackers with a low IoU threshold usually produce noisy samples, ultimately leading to non-convergence in the training phase. It is natural to conclude: the classifier tends to degrade the performance trained with a certain IoU threshold.

We analyze that labels should be assigned primarily according to *gt* rather than the prior anchor box considering the distribution of the target and background for each search image in the training data is different. Thus, the IoU threshold should adjust dynamically to accommodate the characteristics of the search images. In this study, we propose a novel dynamic label assignment (DLA) module based on target statistical properties to fully fit the object tracking community referring to ATSS [29] in object detection. This approach can adaptively define the dynamic IoU threshold and reasonably select positive/negative training samples. First, we apply the statistical sample label assignment approach to the field of object tracking. Additionally, to verify the effectiveness of the proposed algorithm, we integrated DLA into two representatives anchor-based SiamRPN++ [18] and anchor-free SiamCAR [26]. The major discrepancy with ATSS [29] is concentrated on the definition of statistical parameters caused by different numbers of *gt* and feature pyramid levels. Fig. 1 shows the existing scale or aspect ratio handling approaches and the corresponding label assignment results. Extensive experiments conducted on multiple object-tracking datasets show that our approach can bring performance gains in both SiamRPN++ [18] and SiamCAR [26]. Especially, the proposed algorithm does not introduce any extra overhead to the inference phase, and hence, the speed can remain consistent with the original trackers.

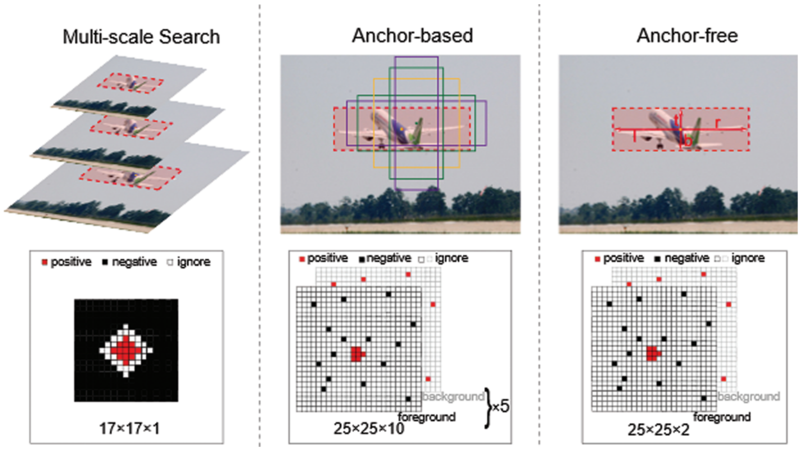


Figure 1: Popular scale and aspect ratio handing approaches and corresponding label assignment results of multi-scale search, anchor-based and anchor-free. (For multi-scale search, 🟥positive represents positive sample labels; while for anchor-based and anchor-free, 🟥positive in the foreground and ■negative in the background are shown as positive sample labels, ■negative in the foreground and 🟥positive in the background are shown as negative sample labels, and □ignore in the foreground and □ignore in the background are indicated as ignore sample labels.)

The primary contributions can be summarized in the following three folds.

1. We propose a novel DLA module to automatically define positive/negative training samples based on the statistical characteristics of the target.
2. The DLA module is embedded in anchor-based and anchor-free siamese trackers with simple, intuitive, and convenient designs. The only hyperparameter involved in DLA is proved to be quite robust.
3. Extensive experiments on public benchmarks show that our DLA module can enhance the performance of trackers without decreasing the inference speed.

2 Related Work

Section 2.1 discusses the characteristics, categories and developments from multi-perspectives of Siamese-based object tracking. Then we state the definition and importance of label assignment, review current fixed label assignment strategies in object tracking community, and present attempts at dynamic label assignment in object detection that may inspire our work in Section 2.2.

2.1 Siamese-Based Object Tracking

Over the past few years, current Siamese-based object trackers, leading to the co-occurrence of superior accuracy and high speed, have achieved top-notch advances. Early Siamese-based trackers resorted to multi-scale search [10] or sampling-then-regression strategies [11] to estimate the target state. The former searches multi-scale regions in a single forward pass and then determines the object size as the size level where the maximum response is located. The latter generates some bounding boxes and selects the best one to regress more precise results. However, these approaches are fundamentally limited because target state estimation is a complex task that requires high-level knowledge of targets. Subsequently, many attempts have been made to refine the target state estimation. Particularly, anchor-based trackers replace the traditional multi-scale search approach due to superior accuracy-speed trade-off. Both pipelines perform a common paradigm: designing a classifier and regressor to solve the location and regression tasks. Several trackers revolve around the two pillars since the seminal work SiamRPN [16] was proposed. They enhance the performance of classifiers from various standpoints, for example, replacing deeper backbone networks [23], adding attention mechanisms [22,31,32], introducing a multi-stage tracker with a cascade architecture [21], emphasizing hard negative samples, and so on [17,28,33,34]. However, the massive preset anchors rely on heuristic knowledge and manual tuning of hyperparameters about the anchors. Thus, recently, elegant-yet-effective anchor-free trackers [24–28] are proposed to address the issues caused by the use of anchors.

2.2 Training Sample Label Assignment

Label assignment is the task of defining a candidate anchor (or a grid cell in anchor-free models) as a positive or negative sample. How to assign labels has been identified as a crucial factor that deeply affects the classifier's performance. We divide mainstream label assignment approaches into two major categories: fixed and dynamic. Fixed label assignment can also be roughly divided into Distance-based and IoU-based. Distance-based label assignment usually takes the distance from the center of gt as the standard but is slightly different in the implementation details. For example, in SiamFC [10], the origin o shows the center point of gt; (i, j) is any point in the response map; R is the L2 (Euclidean) distance between (i, j) and o; R_1 and R_2 $(R_1 < R_2)$ is the specific value of R. The points in the response map can be divided into three regions: I $(0 \leq R \leq R_1)$, II $(R_1 < R \leq R_2)$, and III $(R > R_2)$, named positive, ignored, and negative samples, respectively. The positive sample region is shaped like a circle. The set of positive samples in SiamCAR [26] is shaped like a rectangle and the ignored samples are not set. SiamBAN [25] takes the object scale and aspect ratio into account and refines the label assignment strategy using the ellipse shape of positive samples. IoU-based label assignment dates back to faster-RCNN [19] and has been employed in most trackers recently.

As the fixed label assignment strategies may cause suboptimal solutions for the various target distributions, modern detectors are inclined to adopt dynamic label assignment employing adaptative mechanisms. MetaAnchor [35] formulates anchor functions dynamically generated from the randomly customized prior boxes thanks to weight prediction. GuidedAnchoring [36] introduces semantic features to sparse and nonuniform anchoring schemes. Except for the redesign of the anchor prior, some studies [37–39] propose a soft-weighed sample approach for each anchor point. Additionally, some studies [40–42] try seeing label assignments from different viewpoints. PAA [40] models the anchor assignment as a maximum likelihood estimation based on the probability distribution. OTA [41] formulates the label assignment procedure as an Optimal Transport problem, which has been well-studied in optimization theory. PISA [42] ranks the examples based on their desired property (IoU or classification score). This work demonstrates, if balanced properly, prime samples are more useful for training compared to hard samples. MAL [43] selects the positive samples by jointly optimizing their localization and classification scores. Zhang et al. [44] design a mutual labeling approach to reduce the divergence between localization and classification. AutoAssign [45] presents a confidence weighting module for the automatic assignment of each instance. ATSS [29] selects high-quality positive samples by the dynamic IoU threshold. Above all, the detectors show that a proper dynamic label assignment strategy plays a crucial role in algorithm performance, but there are still few relevant studies in the object tracking field. Therefore, we elaborate on how the label assignment affects classification accuracy. Then, we propose a DLA module and apply it to existing anchor-based and anchor-free trackers.

3 Method

In this section, we develop a DLA module for tracking to better use the dynamic characteristics of the training procedure. DLA is implemented on the anchor-based tracking algorithm SiamRPN++ [18] and anchor-free counterpart SiamCAR [26]. Our noteworthy insight is to adaptively select the optimal IoU threshold based on the statistical properties of instances. In the following, we first briefly revisit SiamRPN++ [18] and SiamCAR [26] on their original label assignment strategies. Then, we show how DLA upgrades the performance of original algorithms by designing a novel positive/negative training sample selection mechanism.

3.1 DLA in Anchor-Based Tracking

Fig. 2 shows the network architecture of the proposed DLA-SiamRPN++. The original SiamRPN++ [18] feeds a pair of images into the siamese network and outputs a classification (cls) map and a regression (reg) map. Meanwhile, the algorithm also produces a set of anchor boxes with different aspect ratios to predict the labels of potential targets through the mentioned cls branch and estimate coordinates of potential targets with the reg branch. In the training phase, we should first define positive and negative training samples and then use the positive training samples for regression. As for the original SiamRPN++ [18], the approach to identify positive and negative samples follows Faster R-CNN [19], labels serving as supervisory signals are assigned according to anchors overlapping with *gt*. Formally, the paradigm can be formulated in the following way:

$$label = \begin{cases} 1, & if\ IoU > th_{high} \\ 0, & if\ IoU < th_{low} \\ -1, & otherwise \end{cases} \tag{1}$$

where IoU is the matching degree between *gt* and anchors. th_{high} and th_{low} represent the constant thresholds, which are typically set to 0.6 and 0.3. Positive, negative, and ignored samples are labeled as 1,0 and -1. Additionally, one training pair selected 64 samples, of which up to 16 were positive samples.

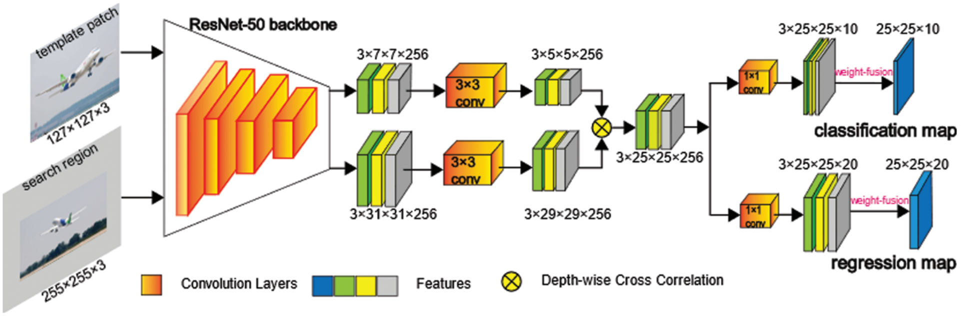


Figure 2: Network architecture of the proposed DLA-SiamRPN++

Given our previous analysis, the heuristic approach to picking positive and negative samples may not be appropriate as the IoU thresholds are empirically selected and fixed for all targets, regardless of their scale and aspect ratios. Therefore, we introduce a DLA module to mitigate the problem of hand-picked priors. Algorithm 1 describes the completed procedure of DLA. The detailed steps are as follows:

Step 1: $R_g\left(x_g, y_g\right)$ and $R_a\left(x_a, y_a\right)$ show the center point coordinates of *gt* and random anchors, respectively. $d\left(R_g, R_a\right)$ represents the L2 distance between the two points.

$$d\left(R_g, R_a\right) = \sqrt{\left(x_a - x_g\right)^2 + \left(y_a - y_g\right)^2} \tag{2}$$

We select top-$\mathcal{K}$ anchor points from $\mathcal{A}$ with the lowest $d\left(R_g, R_a\right)$. There are $w \cdot h$ anchor points on the classification feature map $A^{cls}_{w \cdot h \cdot m}$ and each anchor point tiles m anchor boxes with different aspect ratios $\{1{:}\ 3, 1{:}\ 2, 1{:}\ 1, 2{:}\ 1, 3{:}\ 1\}$. The total number of selected anchor boxes is $m\mathcal{K}$. Previous studies [29,40,41] have sufficiently demonstrated that $\mathcal{K}$ is an insensitive hyperparameter.

Step 2: Compute the IoU between the selected anchors and *gt* as I_g in Line 5.

Step 3: Compute the mean and standard deviation of the IoU results I_g as M_g and V_g in Lines 6 and 7.

Algorithm 1: Dynamic label assignment (DLA) in SiamRPN++

```
Input:
    $\mathcal{I}$ is an input image
    $\mathcal{A}$ is a set of anchors
    $\mathcal{G}$ is the gt annotation for the object within $\mathcal{I}$
    $\mathcal{K}$ is a hyperparameter with a default of 9
Output:
    $\mathcal{P}$ is a set of positive samples
    $\mathcal{N}$ is a set of negative samples
    $\Im$ is a set of ignored samples
1: $\mathcal{P}, \mathcal{N}, \Im \leftarrow \varnothing$;
2: Build an empty set $P_g$ for recording the candidate-positive samples of the gt: $P_g \leftarrow \varnothing$;
```

(Continued)

Algorithm 1: Continued

```
3:  $S \leftarrow$ Select top-$m\mathcal{K}$ anchors from $\mathcal{A}$ with lowest $d\left(R_g, R_a\right)$; $\omega$ Eq. (2)
4:  $P_g = S \cup P_g$;
5:  Compute IoU between $P_g$ with $gt$: $I_g = IoU\left(P_g, gt\right)$;
6:  Compute mean of $I_g$: $M_g = Mean\left(I_g\right)$;
7:  Compute the standard deviation of $I_g$: $V_g = Std\left(I_g\right)$;
8:  Compute  IoU threshold for $gt$: $T_g = M_g + V_g$;
9:  for each selected anchor $p \in P_g \sim$ do
10:     if $IoU\,(p, gt) \geq T_g$ and the center of $p$ in $gt$ then
11:         $\mathcal{P} = \mathcal{P} \cup p$;
12:     end if
13: end for
14: $num\,(\mathcal{N}) : num\,(\mathcal{P}) = 4 : 1$
15: Random_sample $(\mathcal{N}, \Im)$ and $\mathcal{N} \cup \Im = \mathcal{A} - \mathcal{P}$
16: return $\mathcal{P}, \mathcal{N}, \Im$
```

Step 4: The IoU threshold can be statistically expressed as $T_g = M_g + V_g$ in Line 8. IoU mean M_g shows how well the selected anchors match the *gt*. The quality of the selected anchors is positively associated with the value of M_g. IoU standard deviation is employed to select the most appropriate feature pyramid level to detect the object. However, in object tracking, the feature pyramid network is not applied. We conduct layer-wise feature aggregation to achieve rich hierarchical representations instead. Features output from each layer has the same spatial resolution due to our modification of the ResNet-50 backbone. We obtain the final single feature map after weight fusion. Therefore, we have only one total IoU standard deviation V_g. We use the sum of mean M_g and standard deviation V_g as IoU threshold T_g to depict the statistical characteristics of the target.

Step 5: The top-$m\mathcal{K}$ anchors whose IoU with *gt* are equal or greater than T_g can be identified as positive samples in Lines 9 to 13. Meanwhile, we should also follow the center prior guideline that the center of the positive sample is limited to *gt* in Line 10. We find that forcing trackers focus on center areas of objects can help stabilize the early stage of the training process and lead to superior final performance.

Step 6: The rest of the anchors from $\mathcal{A}$ are randomly composed of negative and ignored samples. To alleviate the sample imbalance, the number of negative samples is usually set to four times the number of positive samples. The cross-entropy loss for classification is

$$\mathcal{L}_{cls} = \frac{1}{w \cdot h \cdot m} \sum_{i=1}^{w} \sum_{j=1}^{h} \sum_{k=1}^{m} \left[y_{(i,j,k)} \cdot log\mathcal{F} + \left(1 - y_{(i,j,k)}\right) \cdot log\mathcal{F} \right] \tag{3}$$

$$\mathcal{F} = A^{cls}_{w \cdot h \cdot m}\,(i, j, k)\,. \tag{4}$$

where $\mathcal{F}$ is the value of any anchor on the classification feature map and $y_{(i,j,k)}$ is the label of that anchor. Positive, negative, and ignored samples are labeled as 1, 0, and −1. Ignored samples are not involved in the computation of the classification loss as described above.

DLA shows the design criteria for the classification labels, and the regression labels follow the spatial transformation relationship between positive samples and *gt*.

3.2 DLA in Anchor-free Tracking

Fig. 3 shows the network architecture of the proposed DLA-SiamCAR. SiamCAR [26] primarily has two innovations in network architecture. It is generally appreciated that reasoning with fused layers of deep neural networks can capture rich hierarchical features, where low layers retain fine-grained visual attributes that are robust to precise localization and high layers encode semantic information that is indispensable for discrimination. Both SiamRPN++ [18] and SiamCAR [26] use multiple features extracted from the last three residual blocks of ResNet-50. In contrast to SiamRPN++ [18] feeding the three extracted features into the RPN head and performing layer-wise aggregation with a weighted sum strategy after learning classifiers and regressors individually, SiamCAR [26] concatenates the three extracted features as a unity immediately and then the fused feature map is fed into a classification-regression head after the dimension-reduction operation to predict objects. The adjustment of SiamCAR [26] achieves more consistent operations with conventional CNN, which divides the network into two parts, the backbone to extract features and the head to solve specific tasks. Through the adjustment, the number of parameters is greatly reduced; thus, the computation can be accelerated. Another improvement is that SiamCAR [26] adds a center-ness branch in parallel with the classification branch to filter out the low-quality bounding boxes. The center-ness branch describes the normalized distance between the location and the center of the target, reducing the weights of the bounding boxes away from the target.

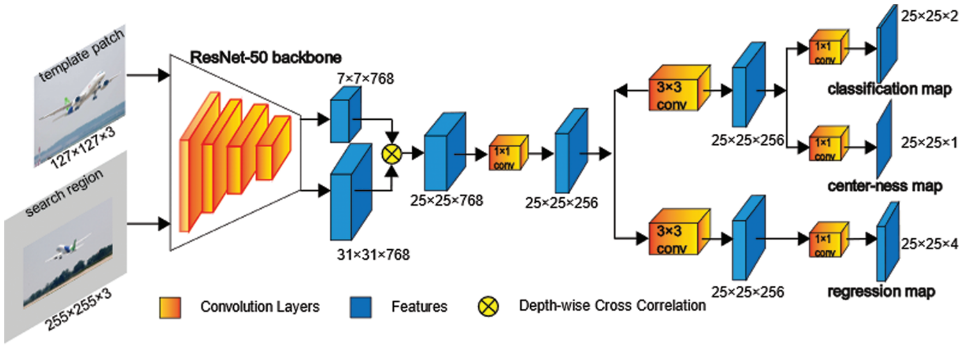


Figure 3: Network architecture of the proposed DLA-SiamCAR

The original SiamCAR [26] uses a distance-based label assignment approach, and the details are as follows. Each element in the classification map can be mapped back to the input search region, and each search region marks a *gt* for the target. The width, height, and center point of the *gt* are represented by $w_g, h_g, \left(x_{gc}, y_{gc}\right)$. With the center, width, and height are $\left(x_{gc}, y_{gc}\right)$, $0.6w_g$, and $0.6h_g$, we can generate a rectangle *Re*. They consider elements of the classification map within the region *Re* to be positive samples. The rest elements that fall outside the region *Re* are assigned with negative labels. No buffer is set to ignore some ambiguous samples. As stated earlier, anchor-based trackers regard the element as the center of multiple anchor boxes with different aspect ratios, and these anchor boxes regress the target bounding boxes as references. Unlike them, SiamCAR [26] belongs to the anchor-free tracker, which directly regresses to the target bounding box on the element. The regression map is a 4D real vector that includes the distances from the center of the search region to the four sides of *gt*. We argue that this approach can partly reflect the size of the target, but there is still a certain gap with the dynamic label assignment strategy in terms of adaptively defining the positive/negative training samples.

Thus, we introduce a DLA module similar to Section 3.1 and modify it to fit the anchor-free algorithm SiamCAR [26]. The top-$\mathcal{K}$ anchor points are selected according to their closest L2 distance to the center of gt. Tile an anchor box at each selected anchor point. A previous study [29] demonstrated that anchor setting is robust to aspect ratios, and hence we chose the simple and representative anchor boxes with the size 64 × 64 (scale = 8, stride = 8, aspect ratio = 1:1). After that, we compute the IoU between the selected $\mathcal{K}$ anchor boxes and gt, and further compute IoU mean and standard deviation. The IoU threshold is the sum of the mean and standard deviation. Finally, the anchor points whose corresponding anchor boxes IoU are greater than or equal to the IoU threshold are defined as positive samples. Notably, the positive samples were limited to gt. The remaining anchor points are randomly defined as negative samples or ignored samples and restrict the number of negative samples to four times positive samples. Positive, negative, and ignored samples are labeled as 1, 0 and -1.

4 Experiments

4.1 Implementation Details

Training. We implement the proposed approach using PyTorch on a server with Intel(R) Xeon(R) Gold 6246 CPU @ 3.30 GHz and four NVIDIA RTX 2080ti GPUs with CUDA 10. The implementation details for our DLA-SiamRPN++ and DLA-SiamCAR algorithms remain consistent. Since the model requires pair-wise inputs, we randomly pick two images from the sequences containing the same object and extract the template patch and search region of 127 × 127 and 255 × 255 pixels respectively. The image pairs are normalized by the per-color mean ($mean = [0.485, 0.456, 0.406]$) and standard deviation ($std = [0.229, 0.224, 0.225]$) according to the practice of [46]. Next, we also apply data augmentation operations, including random translations, blur, stretch and color jitter. Processing-completed image pairs are known as training samples.

Network architecture. We adopt the modified ResNet-50 [46] as our backbone and initialize the pre-trained parameters on ImageNet during the training phase [47]. For other parameters, the weights are initialized with Kaiming algorithm [48] and the biases are initialized to zero. We optimize the final parameters by SGD. The large-scale public training datasets of COCO [49], VID [47], DET [47], LaSOT [50], GOT-10k [51], YouTube-BB [52] and TrackingNet [53] are used to train the network end-to-end, which yield 6×10^5 image pairs for one epoch. There are 20 epochs in total. We optimize the final parameters by SGD and employ the learning rate with a warm-up. Notably, the weights of the backbone network are frozen until the 11th epoch.

Evaluation. After data augment, the first frame of the video with gt is processed as a standard template patch, which remains fixed throughout the inference phase. We adopt the same online tracking strategy as SiamRPN++ [18] in the inference phase. We evaluate our algorithms using current state-of-the-art trackers on object tracking benchmarks OTB100 [54], UAV123 [55], VOT2019 [56] and LaSOT [50], following corresponding protocols for experiments. All the employed datasets are summarized in Table 1.

Table 1: A summary of the employed datasets in terms of type, scale and other features

Dataset	Type	Train	Test	Object instances	Object classes	Frame rate
COCO [49]	Image	123K	/	0.5M	80	/
ImageNet DET [47]	Image	456K	/	0.47M	200	/
ImageNet VID [47]	Video	5.4K	/	2.7M	30	30
LaSOT [50]	Video	1.12K	280	3.3M	70	30
GOT-10k [51]	Video	9.34K	/	1.5M	563	10
YouTube-BB [52]	Video	240K	/	5.6M	23	1
TrackingNet [53]	Video	30K	/	14M	27	30
OTB100 [54]	Video	/	100	59K	16	30
UAV123 [55]	Video	/	123	113K	9	30
VOT2019 [56]	Video	/	60	19.9K	30	30

4.2 Results on OTB100

Performance Index. The OTB100 [54] public dataset is a widely employed classical benchmark for evaluating the performance of single object trackers and contains 100 challenging sequences with significant variations. We use a standard one-pass evaluation success rate and precision as evaluation metrics. We simultaneously compute the IoU between the predicted bounding box and *gt* and the distance of their central locations for each frame. As illustrated in Fig. 4, we obtain the success plot by computing the success rate at a threshold varying from zero to one and measure the final success score by the area-under-curve (AUC) of the plot, and the precision plot is obtained similarly and measures the threshold at 20 pixels as the final precision score. We compare our DLA-SiamRPN++ and DLA-SiamCAR with several state-of-the-art trackers, i.e., SiamRPN++ [18], SiamRCNN [57], SiamCAR [26], RBO [58], SiamFC++ [24], DaSiamRPN [17], ATOM [59] and SAOT [60]. DLA-SiamRPN++ can rank 2nd, especially outperforming the baseline SiamRPN++ [18] with relative improvements of 1% in success and 0.5% in precision. Furthermore, DLA-SiamCAR outperforms the baseline SiamCAR [26] with relative improvements of 2.5% in success and 0.6% in precision. This shows that the DLA module has effective improvement on their corresponding top-performing algorithms.

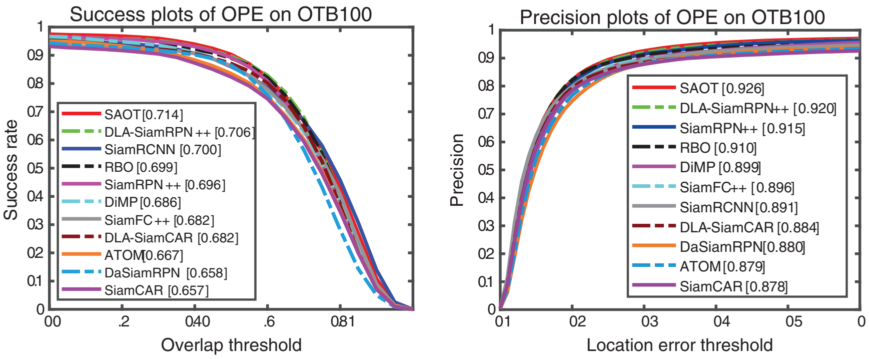


Figure 4: Success and precision plots on the OTB100 dataset

To further validate the effectiveness and generality of the DLA module, we perform more comparative experiments by exploring two new baselines, CoSiamRPN [61] (an anchor-based tracker with EfficientNet-B0 [62] backbone) and SiamFC++ [24] (an anchor-free tracker with GoogLeNet [63] backbone). The results of OTB100 [54] dataset in terms of the success AUC score and precision score are shown in Table 2. We observe that our proposed DLA module achieves consistent AUC improvements on all trackers with different backbones, which demonstrates its universality.

Table 2: Results of different trackers on OTB100 dataset

Method	Backbone	AUC	Prec
Anchor-based trackers			
SiamRPN++ [18]	ResNet-50	0.696	0.915
CoSiamRPN [61]	EfficientNet-B0	0.684	/
DLA-SiamRPN++	ResNet-50	**0.706 (+1%)**	0.920
DLA-CoSiamRPN	EfficientNet-B0	**0.687 (+0.3%)**	0.899
Anchor-free trackers			
SiamCAR [26]	ResNet-50	0.657	0.878
SiamFC++ [24]	GoogLeNet	0.682	0.896
DLA-SiamCAR	ResNet-50	**0.682 (+2.5%)**	0.884
DLA-SiamFC++	GoogLeNet	**0.684 (+0.2%)**	0.896

Attribute Analysis. To address the different aspect challenges in the tracking process, all test videos in OTB100 [54] are manually labeled with 11 specific attributes, i.e., Illumination Variation (IV), Scale Variation (SV), In-Plane Rotation (IPR), Out-of-Plane Rotation (OPR), Background Clutters (BC), Deformation (DEF), Occlusion (OCC), Fast Motion (FM), Motion Blur (MB), Low Resolution (LR), and Out-of-View (OV). As shown in Fig. 5, we further evaluate the success rate of the above algorithms under all 11 specific attributes to verify the performances of our algorithms in detail. Success rate is the most important metric, the greater the success rate, the better the performance. In almost all challenging scenarios, the proposed DLA-SiamRPN++ outperforms its baseline SiamRPN++ [18] in almost all of the attributes, and the DLA-SiamCAR is also much better than the corresponding baseline SiamCAR [26]. DLA-SiamRPN++ and SiamRPN++ [18] can better handle the challenges except for LR and OV than DLA-SiamCAR and SiamCAR [26]. This is because variations in target appearance account for a large proportion of these video sequences. The rest nine attributes are more prone to variations in target shape and pose. The phenomenon suggests that DLA-SiamRPN++ and DLA-SiamCAR have their strength in addressing the challenges of object tracking.

Qualitative Evaluation. We visualize the tracking results on certain sequences from OTB100 [54] due to the limited length of the paper to qualitatively showcase the performances of the proposed approaches. Fig. 6 visually compares the findings of other recent state-of-the-art trackers: ATOM [59], SiamRCNN [57], SiamRPN++ [18], and SiamCAR [26]. The numbers in the top left corner of the legend show the order of the sequence frames. Furthermore, corresponding video results are provided at video URLs to check the quality of the trackers over time.

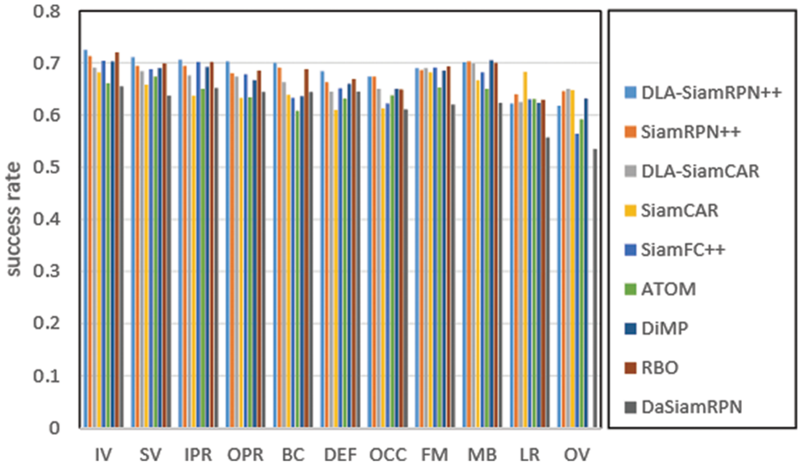
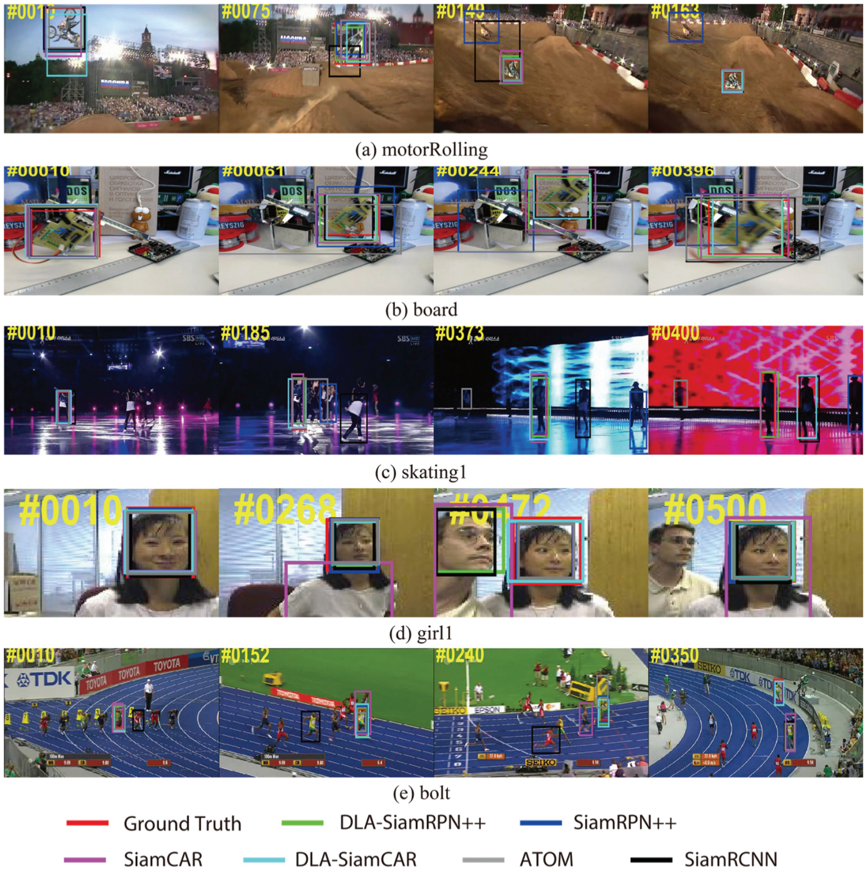


Figure 5: Particular attributes comparison of the trackers on the OTB100 dataset in terms of success plot AUC



(a) motorRolling

(b) board

(c) skating1

(d) girl1

(e) bolt

Figure 6: Visualization results comparing our methods with their baseline and other top-performing trackers

From the cases of Figs. 6a–6c, the baseline SiamRPN++ [18] tends to drift from the object when similar background clutters are represented around the object. However, the proposed DLA-SiamRPN++ can closely keep up with the objects facing the same scenarios. DLA-SiamCAR successfully follows the object while SiamCAR [26] fails, including the very challenging sequences illustrated in Figs. 6d and 6e. It is verified that DLA-SiamCAR [26] can further enhance the overall performance of SiamCAR [26]. These findings validate the superior tracking effectiveness of our proposed DLA modules as supplements to Figs. 4 and 5.

A plausible explanation is that the label assignment approaches of SiamRPN++ [18] and SiamCAR [26] typically cause them to mistakenly define adjacent pixels as positive samples, thus, identifying the distractor as the same object. Comparatively, the DLA module may enrich global information to generate robust appearance features and adaptively define positive and negative samples customized for each object and achieve precise localization.

4.3 Results on UAV123

UAV123 [55] contains 123 low-altitude aerial sequences with more than 110K frames. Inherently different from the current popular tracking benchmark, UAV123 [55] features fast motion, occlusion, illumination variations, and small objects. This benchmark is designed for UAVs, which are a major application of tracking algorithms. We compare DLA-SiamCAR and DLA-SiamRPN++ with other four recent prevailing trackers, including DaSiamRPN [17], SiamBAN [25], SiamCAR [26], SiamRPN++[18], SiamRPN [16], ECO [64] and DiMP [65] as shown in Table 3. Specifically, our DLA-SiamRPN++ achieves an AUC success score of 0.641 and a precision score of 0.847, suppressing baseline SiamRPN++ [18] by a significant margin. DLA-SiamCAR outperforms its baseline SiamCAR [26] on both success rate and precision.

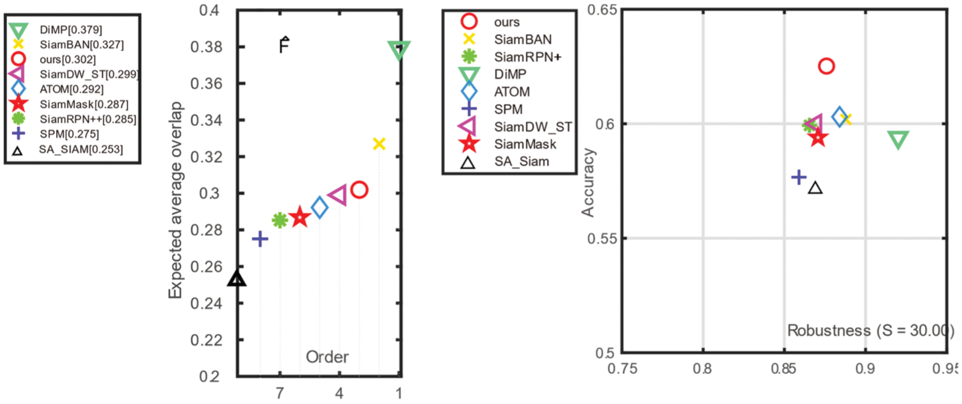
Table 3: Success and precision comparison with state-of-the-art trackers on UAV123. The DLA-enhanced and baseline results are highlighted in red and blue fonts, respectively

Trackers	Success rate	Precision
DaSiamRPN [17]	0.586	0.796
SiamBAN [25]	0.631	0.833
SiamCAR [26]	0.614	0.760
DLA-SiamCAR	0.624	0.833
SiamRPN++ [18]	0.613	0.807
DLA-SiamRPN++	0.641	0.847
SiamRPN [16]	0.557	0.710
ECO [64]	0.525	0.688
DiMP [65]	0.643	0.791

4.4 Results on VOT2019

Next, we evaluate the 2019 version of the Visual Object Tracking challenge (VOT2019 [56]), consisting of 60 public sequences. We employ accuracy, robustness, and Expected Average Overlap (EAO) as metrics. Accuracy measures the average IoU between the predicting bounding boxes and *gt*.

Robustness is the number of tracking failures. EAO can be seen as a primary measure that comprehensively combines the above two metrics in a principled way. Trackers are restarted at failure following the evaluation protocol of VOT2019. Fig. 7a illuminates the EAO rank of eight trackers: our DLA-SiamRPN++, SiamRPN++ [18], SiamMask [66], SiamBAN [25], DiMP [65], ATOM [59], SPM [20], SA_Siam [15] and SiamDW_ST [23]. Based on the value of EAO, the tracker in the right-most has the highest performance. Our approach ranks third and exceeds by 5.9% compared with the baseline SiamRPN++ [18]. DLA-SiamRPN++ achieves the highest accuracy but only gets the third robustness, as illustrated in Fig. 7b. The value of robustness in Fig. 7b undergoes mathematical processing for better presentation by $exp\,(-0.01 \times S \times R)$ where S is the number of frames continuously tracked and R is the tracking failure rate. Particularly, our approach DLA-SiamRPN++ outperforms the baseline SiamRPN++ [18] in terms of accuracy, robustness, and EAO, thus, demonstrating the effectiveness and efficiency of the introduced DLA module.



(a) (b)

Figure 7: Comparison with state-of-the-art trackers on the VOT2019 dataset. (a) Expected average overlap performance ranking; (b) A-R plot for the experiment baseline generated by sequence pooling

4.5 Results on LaSOT

LaSOT [50] is a large-scale, high-quality, dense-annotation, and long-term dataset with 1400 sequences, 70 categories, and 3.5 million frames. Additionally, the benchmark sets up plenty of challenges from the wild such as the object can be out-of-frame and then reappear in the view, so the evaluation platform is convincing and truly shows the performance of trackers. We trained our approach on the LaSOT [50] training subset and tested it on the LaSOT [50] test subset. Success rate and normalized precision are used as evaluation metrics. Fig. 8 shows the overall performance of trackers. Our DLA-SiamRPN++ ranks top both in success rate and precision, further validating that our DLA module can realize superior performance in long-time tracking even without the model online update.

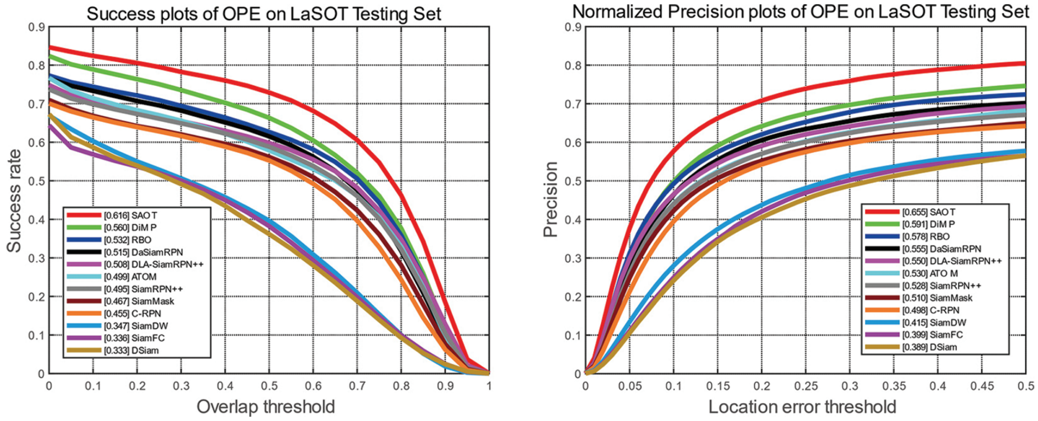
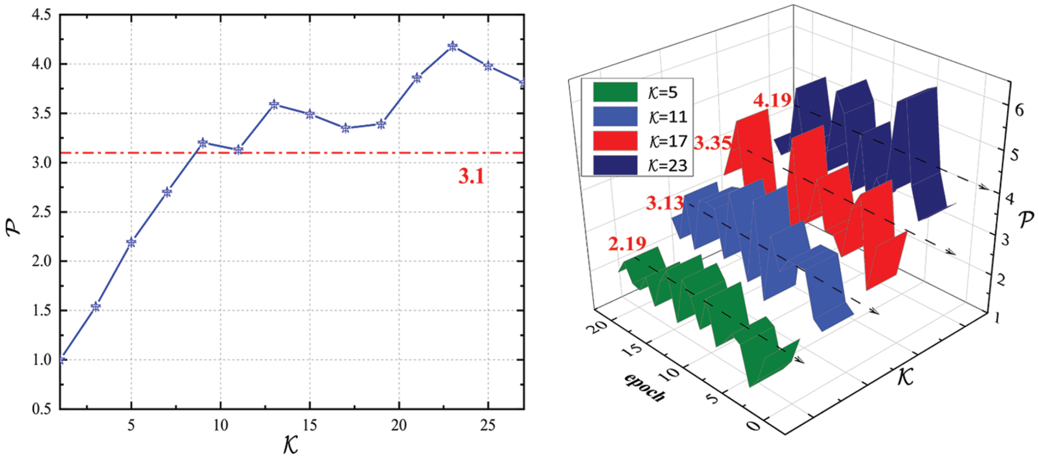


Figure 8: Success and precision plots on the LaSOT dataset

4.6 *Hyper-Parameter Analysis*

We conducted a series of experiments to investigate the robustness of the only hyperparameter $\mathcal{K}$ w.r.t DLA. We use different values of $\mathcal{K}$ to train the tracker, as illustrated in Fig. 9a. The vertical coordinates $\mathcal{P}$ represent the mean of the statistical number of positive samples after 20 epochs of training (with 600,000 sequences per epoch). Experimental findings reveal that the values of $\mathcal{P}$ range from 3 to 4. This shows that the number of positive samples can remain stable even if the magnitude of $\mathcal{K}$ change is large. Fig. 9b illustrates the variations of $\mathcal{P}$ at different epochs when $\mathcal{K} = 5, 11, 17, 23$. The findings show that the value of $\mathcal{P}$ changes gently with constant $\mathcal{K}$. In summary, DLA-SiamCAR is quite insensitive to $\mathcal{K}$ from 7 to 19 and then we can conclude that the proposed DLA module can be nearly seen as hyperparameter-free.



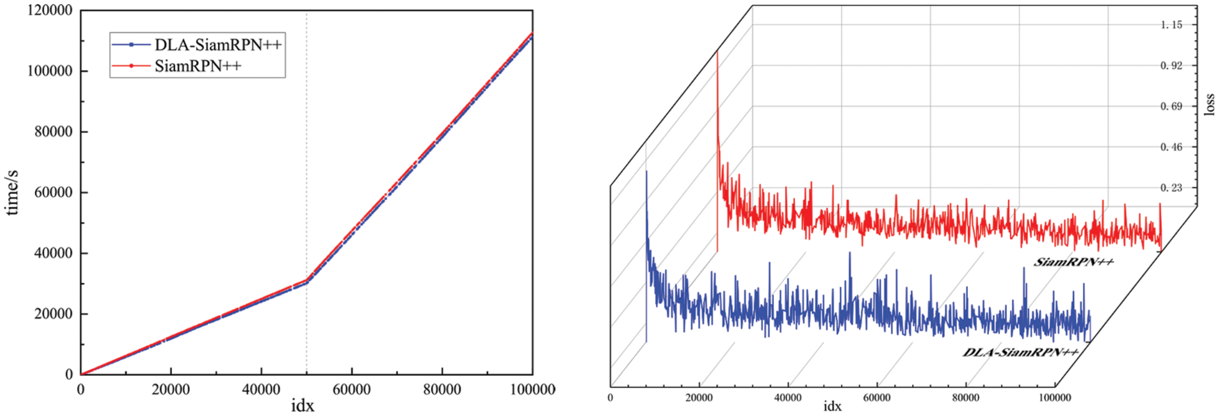
(a)

(b)

Figure 9: (a) Analysis of different $\mathcal{K}$ on training datasets. $\mathcal{P}$ shows the mean of the statistical number of positive samples after 20 epochs of training (with 600,000 sequences per epoch). (b) Variation in the number of selected positive samples vs. the epoch when $\mathcal{K} = 5, 11, 17, 23$. The figures illustrate that the proposed DLA module can be nearly seen as hyperparameter-free

4.7 Speed Comparison Analysis

Training speed comparison. We evaluate the training cost of the DLA module from two perspectives: training time and training loss. The comparative results between DLA-SiamRPN++ and its baseline SiamRPN++ [18] are shown in Fig. 10. There are 20 epochs in total. Each epoch hosts 6×10^5 training pairs.



(a) (b)

Figure 10: Results for training cost comparison between DLA-SiamRPN++ and SiamRPN++

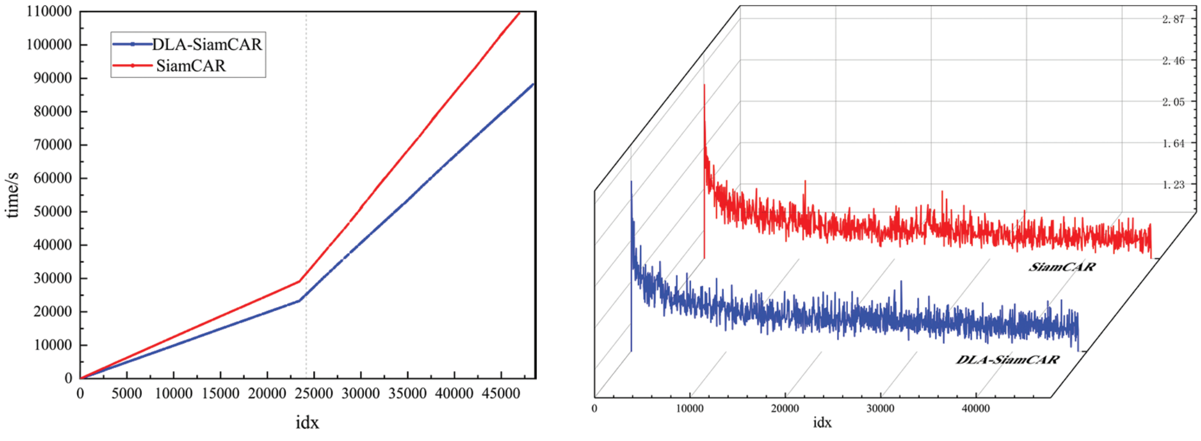
We set the mini-batch size to 120 so that one epoch enquires 5000 iterations.

As shown in Fig. 10a, the training time varies linearly with iterations. Specifically, the inflection point (iteration $=5 \times 10^4$) means the training speed has changed at the 11th epoch. The reason for the speed change is that the backbone network parameters are frozen for 10 epochs before the backbone joins the training at the 11th epoch. For SiamRPN++ [18], 0.624 s per iteration from epoch 0 to 10 and 1.63 s per iteration from epoch 11 to 20. For DLA-SiamRPN++, 0.60 s per iteration from epoch 0 to 10 and 1.62 s per iteration from epoch 11 to 20. This suggests DLA module can slightly improve the training speed of SiamRPN++ [18].

Similarly, we present the comparative results between DLA-SiamCAR and its baseline SiamCAR [26] in Fig. 11. Notably, the model parameter of SiamCAR [26] is 5.58 M, which is less than 17.25 M of SiamRPN++ [18]. We set the mini-batch size to 256 so that one epoch enquires 2343 iterations. For SiamCAR [26], 1.24 s per iteration from epoch 0 to 10 and 3.33 s per iteration from epoch 11 to 20. For DLA-SiamCAR, 0.99 s per iteration from epoch 0 to 10 and 2.61 s per iteration from epoch 11 to 20. DLA module achieves large speed improvements on SiamCAR [26], indicating that DLA module is more effective than the fixed label assignment strategy.

DLA module is a two-step selection mechanism. The first step selects candidate samples according to the Euclidean distance between training samples and *gt*, and the second step selects positive/negative samples based on the mean and standard deviation of candidate samples. Although this process requires additional computation, the number of positive samples involved in the loss computation is refined, ultimately reducing the training cost.

Figs. 10b and 11b show no significant differences in the training loss trajectories after the introduction of the DLA module.



(a) (b)

Figure 11: Results for training cost comparison between DLA-SiamCAR and SiamCAR

Inference speed analysis. We state that the speed of the object tracking algorithm with the DLA module can be consistent with the original algorithm. Fundamentally, the DLA module that helps implement a superior model will only affect the training phase. Next, we load the trained model to test the performance of the object tracking algorithm. Since the algorithm framework is the same in the inference phase and the values of the loaded model parameters are the only difference, so the test speed is naturally the same as the original algorithm.

5 Conclusion

In this study, we propose a novel dynamic label assignment module, called DLA, to adaptively select positive and negative training samples based on the statistical characteristics of objects. Then, we apply the DLA module to anchor-based and anchor-free trackers. Extensive experiments show that the DLA module can enhance the performance of object trackers and maintains the same tracking speed as the original algorithms. Meanwhile, we experimentally proved that the number of positive samples ultimately selected by the training model is maintained at about three at different $\mathcal{K}$ values. To summarize, we can state with certainty that DLA is simple, effective, and hyperparameter-free. Thus, how to apply a more label assignment approach in the object-tracking community should be further investigated in the future.

Funding Statement: The support of the National Natural Science Foundation of China (Grant No. 52127809, author Z. W, http://www.nsfc.gov.cn/ ; No. 51625501, author Z. W, http://www.nsfc.gov.cn/) is greatly appreciated.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding this study.

References

[1] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan and S. Kasaei, "Deep learning for visual tracking: A comprehensive survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 3943–3968, 2022.

[2] Z. Wang, J. Qin, X. Xiang, Y. Tan and N. N. Xiong, “Criss-cross attentional siamese networks for object tracking,” *Computers, Materials & Continua*, vol. 73, no. 2, pp. 2931–2946, 2022.

[3] Z. Tian, C. Shen, H. Chen and T. He, “FCOS: Fully convolutional one-stage object detection,” in *Proc. ICCV*, Seoul, Korea, pp. 9627–9636, 2019.

[4] N. Dilshad, A. Ullah, J. Kim and J. Seo, “LocateUAV: Unmanned aerial vehicle location estimation via contextual analysis in an IoT environment,” *IEEE Internet of Things Journal*, 2022. https://10.1109/JIOT.2022.3162300.

[5] N. Dilshad, J. Hwang, J. Song and N. Sung, “Applications and challenges in video surveillance via drone: A brief survey,” in *Proc. ICTC*, Jeju Island, SK, pp. 728–732, 2020.

[6] Y. Cho, W. J. Kim, S. Hong and S. Yoon, “Part-based pseudo label refinement for unsupervised person re-identification,” in *Proc. CVPR*, New Orleans, LA, USA, pp. 7298–7308, 2022.

[7] Z. Lu, R. Lin, X. Lou, L. Zheng and H. Hu, “Identity-unrelated information decoupling model for vehicle re-identification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19001–19015, 2022.

[8] D. Held, S. Thrun and S. Savarese, “Learning to track at 100 FPS with deep regression networks,” in *Proc. ECCV*, Amsterdam, The Netherlands, pp. 749–765, 2016.

[9] S. Javed, M. Danelljan, F. S. Khan, M. H. Khan, M. Felsberg *et al.,* “Visual object tracking with discriminative filters and siamese networks: A survey and outlook,” *CoRR*, vol. abs/2112.02838, pp. 1–20, 2021.

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” in *Proc. ECCV*, Amsterdam, The Netherlands, pp. 850–865, 2016.

[11] R. Tao, E. Gavves and A. W. M. Smeulders, “Siamese instance search for tracking,” in *Proc. CVPR*, Las Vegas, NV, USA, pp. 1420–1429, 2016.

[12] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan and M. Felsberg, “Unveiling the power of deep tracking,” in *Proc. ECCV*, Munich, Germany, pp. 483–498, 2018.

[13] Q. Guo, F. Wei, C. Zhou, H. Rui and W. Song, “Learning dynamic siamese network for visual object tracking,” in *Proc. ICCV*, Venice, Italy, pp. 1781–1789, 2017.

[14] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi and P. Torr, “End-to-End representation learning for correlation filter based tracking,” in *Proc. CVPR*, Honolulu, HI, USA, pp. 5000–5008, 2017.

[15] A. He, L. Chong, X. Tian and W. Zeng, “A twofold siamese network for real-time object tracking,” in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 4834–4843, 2018.

[16] L. Bo, J. Yan, W. Wei, Z. Zheng and X. Hu, “High performance visual tracking with siamese region proposal network,” in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 8971–8980, 2018.

[17] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan *et al.,* “Distractor-aware siamese networks for visual object tracking,” in *Proc. ECCV*, Munich, Germany, pp. 103–119, 2018.

[18] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing *et al.,* “SiamRPN++: Evolution of siamese visual tracking with very deep networks,” in *Proc. CVPR*, Long Beach, CA, USA, pp. 4277–4286, 2019.

[19] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[20] G. Wang, C. Luo, Z. Xiong and W. Zeng, “Spm-tracker: Series-parallel matching for real-time visual object tracking,” in *Proc. CVPR*, Long Beach, CA, USA, pp. 3643–3652, 2019.

[21] H. Fan and H. Ling, “Siamese cascaded region proposal networks for real-time visual tracking,” in *Proc. CVPR*, Long Beach, CA, USA, pp. 7944–7953, 2019.

[22] Y. Yu, Y. Xiong, W. Huang and M. R. Scott, “Deformable siamese attention networks for visual object tracking,” in *Proc. CVPR*, Seattle, WA, USA, pp. 6727–6736, 2020.

[23] Z. Zhang and H. Peng, “Deeper and wider siamese networks for real-time visual tracking,” in *Proc. CVPR*, Long Beach, CA, USA, pp. 4586–4595, 2019.

[24] 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*, New York, NY, USA, pp. 12549–12556, 2020.

[25] Z. Chen, B. Zhong, G. Li, S. Zhang and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. CVPR*, Seattle, WA, USA, pp. 6668–6677, 2020.

[26] D. Guo, J. Wang, Y. Cui, Z. Wang and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. CVPR*, Seattle, WA, USA, pp. 6268–6276, 2020.

[27] Z. Zhang, H. Peng, J. Fu, B. Li and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. ECCV*, Glasgow, UK, pp. 711–787, 2020.

[28] Z. Zhang and L. Zhang, "Hard negative samples emphasis tracker without anchors," in *Proc. ACM MM*, Seattle, WA, USA, pp. 4299–4308, 2020.

[29] S. Zhang, C. Chi, Y. Yao, Z. Lei and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. CVPR*, Seattle, WA, USA, pp. 9756–9765, 2020.

[30] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 6154–6162, 2018.

[31] Q. Wang, Z. Teng, J. Xing, J. Gao and S. Maybank, "Learning attentions: Residual attentional siamese network for high performance online visual tracking," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 4854–4863, 2018.

[32] D. Guo, Y. Shao, Y. Cui, Z. Wang and C. Shen, "Graph attention tracking," in *Proc. CVPR*, Online, pp. 9543–9552, 2021.

[33] L. Zhang, A. Gonzalez-Garcia, J. V. D. Weijer, M. Danelljan and F. S. Khan, "Learning the model update for siamese trackers," in *Proc. ICCV*, Seoul, Korea, pp. 4010–4019, 2019.

[34] H. Lee, S. Choi, Y. Kim and C. Kim, "Bilinear siamese networks with background suppression for visual object tracking," in *Proc. BMVA*, Cardiff, Wales, UK, pp. 8, 2019.

[35] T. Yang, X. Zhang, Z. Li, W. Zhang and J. Sun, "MetaAnchor: Learning to detect objects with customized anchors," in *Proc. NIPS*, Montreal, Quebec, Canada, pp. 318–328, 2018.

[36] J. Wang, K. Chen, S. Yang, C. C. Loy and D. Lin, "Region proposal by guided anchoring," in *Proc. CVPR*, Long Beach, CA, USA, pp. 2965–2974, 2019.

[37] H. Song, M. Kim, D. Park, Y. Shin and J. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022. https://10.1109/TNNLS.2022.3152527.

[38] C. Zhu, F. Chen, Z. Shen and M. Savvides, "Soft anchor-point object detection," in *Proc. ECCV*, Glasgow, UK, pp. 91–107, 2020.

[39] Y. Zhai, J. Fu, Y. Lu and H. Li, "Feature selective networks for object detection," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 4139–4147, 2018.

[40] K. Kim and H. S. Lee, "Probabilistic anchor assignment with iou prediction for object detection," in *Proc. ECCV*, Glasgow, UK, pp. 355–371, 2020.

[41] Z. Ge, S. Liu, Z. Li, O. Yoshie and J. Sun, "Ota: Optimal transport assignment for object detection," in *Proc. CVPR*, Online, pp. 303–312, 2021.

[42] Y. Cao, K. Chen, C. C. Loy and D. Lin, "Prime sample attention in object detection," in *Proc. CVPR*, Seattle, WA, USA, pp. 6668–6677, 2020.

[43] W. Ke, T. Zhang, Z. Huang, Q. Ye, J. Liu *et al.,* "Multiple anchor learning for visual object detection," in *Proc. CVPR*, Seattle, WA, USA, pp. 10206–10215, 2020.

[44] T. Zhang, Q. Zhong, S. Pu and D. Xie, "Modulating localization and classification for harmonized object detection," in *Proc. ICME*, Shenzhen, GD, China, pp. 1–6, 2021.

[45] B. Zhu, J. Wang, Z. Jiang, F. Zong, S. Liu *et al.,* "AutoAssign: Differentiable label assignment for dense object detection," *CoRR*, vol. abs/2007.03496, pp. 1–11, 2020.

[46] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, pp. 770–778, 2016.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh *et al.,* "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[48] K. He, X. Zhang, S. Ren and J. Sun, "Delving deep into rectifiers Surpassing Human-Level Performance on ImageNet Classification," in Proc. ICCV, San tiago, Chile, pp. 1026–1034, 2015.

[49] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona *et al.,* "Microsoft coco: Common objects in context," in *Proc. ECCV*, Zurich, Switzerlan, pp. 740–755, 2014.

[50] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng *et al.,* "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. CVPR*, Long Beach, CA, USA, pp. 5369–5378, 2019.

[51] L. Huang, X. Zhao and K. Huang, "GOT-10k: A large High-Diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1562–1577, 2021.

[52] E. Real, J. Shlens, S. Mazzocchi, X. Pan and V. Vanhoucke, "Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video," in *Proc. CVPR*, Honolulu, HI, USA, pp. 5296–5305, 2017.

[53] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. ECCV*, Munich, Germany, pp. 300–317, 2018.

[54] W. Yi, L. Jongwoo and Y. Ming-Hsuan, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[55] M. Mueller, N. Smith and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. ECCV*, Amsterdam, The Netherlands, pp. 445–461, 2016.

[56] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder *et al.,* "The seventh visual object tracking VOT2019 challenge results," in *Proc ICCVW*, Seoul, Korea, pp. 2206–2241, 2019.

[57] P. Voigtlaender, J. Luiten, P. H. S. Torr and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. CVPR*, Seattle, WA, USA, pp. 6577–6587, 2020.

[58] F. Tang and Q. Ling, "Ranking-based siamese visual tracking," in *Proc. CVPR*, New Orleans, LA, USA, pp. 8741–8750, 2022.

[59] M. Danelljan, G. Bhat, F. S. Khan and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. CVPR*, Long Beach, CA, USA, pp. 4660–4669, 2019.

[60] Z. Zhou, W. Pei, X. Li, H. Wang, F. Zheng *et al.,* "Saliency-associated object tracking," in *Proc. ICCV*, Online, pp. 9866–9875, 2021.

[61] K. Tan, T. Xu and Z. Wei, "Learning complementary Siamese networks for real-time high-performance visual tracking," *Journal of Visual Communication and Image Representation*, vol. 80, no. 11, pp. 103299, 2021.

[62] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, Long Beach, CA, USA, pp. 6105–6114, 2019.

[63] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed *et al.,* "Going deeper with convolutions," in *Proc. CVPR*, Boston, MA, USA, pp. 1–9, 2015.

[64] M. Danelljan, G. Bhat, F. S. Khan and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. CVPR*, Honolulu, HI, USA, pp. 6931–6939, 2017.

[65] G. Bhat, M. Danelljan, L. Van Gool and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. ICCV*, Seoul, Korea, pp. 6181–6190, 2019.

[66] Q. Wang, L. Zhang, L. Bertinetto, W. Hu and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. CVPR*, Long Beach, CA, USA, pp. 1328–1338, 2019.