

# Deformable Siamese Attention Networks for Visual Object Tracking

Yuechen Yu, Yilei Xiong, Weilin Huang<sup>†</sup>, Matthew R. Scott  
Malong Technologies

{rogyu, yilxiong, whuang, mscott}@malong.com

## Abstract

*Siamese-based trackers have achieved excellent performance on visual object tracking. However, the target template is not updated online, and the features of the target template and search image are computed independently in a Siamese architecture. In this paper, we propose Deformable Siamese Attention Networks, referred to as SiamAttn, by introducing a new Siamese attention mechanism that computes deformable self-attention and cross-attention. The self-attention learns strong context information via spatial attention, and selectively emphasizes interdependent channel-wise features with channel attention. The cross-attention is capable of aggregating rich contextual interdependencies between the target template and the search image, providing an implicit manner to adaptively update the target template. In addition, we design a region refinement module that computes depth-wise cross correlations between the attentional features for more accurate tracking. We conduct experiments on six benchmarks, where our method achieves new state-of-the-art results, outperforming the strong baseline, SiamRPN++ [24], by  $0.464 \rightarrow 0.537$  and  $0.415 \rightarrow 0.470$  EAO on VOT 2016 and 2018.*

## 1. Introduction

Visual object tracking aims to track a given target object at each frame over a video sequence. It is a fundamental task in computer vision [17, 16, 20], and has numerous practical applications, such as automatic driving [23], human-computer interaction [28], robot sensing, *etc.* Recent efforts have been devoted to improving the performance of visual object trackers. However, developing a fast, accurate and robust tracker is still highly challenging due to the vast amount of deformations, motions and occlusions that often occur on video objects with complex background [38, 22, 10].

Deep learning technologies have significantly advanced the task of visual object tracking, by providing the strong



Figure 1: Tracking results of our deformable Siamese attention networks (SiamAttn) with three state-of-the-art trackers. Our results are more accurate, and are robust to appearance changes, complex background and close distractors with occlusions. Fig. 4 further shows the strong discriminative features learned by our Siamese attention module.

capacity of learning powerful deep features. For example, Bertinetto *et al.* [1] first introduced Siamese networks for visual tracking. Since then, object trackers built on Siamese networks and object detection frameworks have achieved the state-of-the-art performance, such as SiamRPN [25], SiamRPN++ [24], and SiamMask [36]. The Siamese-based trackers formulate the problem of visual object tracking as a matching problem by computing cross-correlation similarities between a target template and a search region, which transforms the tracking problem into finding the target object from an image region by computing the highest visual similarity [1, 25, 24, 36, 44]. Therefore, it casts the tracking problem into a Region Proposal Network (RPN) [13] based detection framework by leveraging Siamese networks, which is the key to boost the performance of recent deep trackers.

Siamese-based trackers are trained completely offline by using massive frame pairs collected from videos, and thus

<sup>†</sup>Corresponding author: whuang@malong.com

the target template can not be updated online. This makes it difficult to precisely track the targets with large appearance variations, significant deformations, or occlusions, which inevitably increase the risk of tracking drift. Furthermore, the convolutional features of the target object and the search image are computed independently in Siamese architecture, where background context information is completely discarded in target features, but is of great importance to distinguish the target from close distractors and complex backgrounds. Recent work attempted to enhance the target representation by integrating the features of previous targets [41, 14], but the discriminative context information from the background is ignored. Alternatively, we introduce a new Siamese attention mechanism that encodes rich background context into the target representation by computing cross-attention in the Siamese networks.

Recently, the attention mechanism was introduced to visual object tracking in [35, 45], which inspired the current work. However, the attentions and deep features of the target template and the search image are computed separately in [35, 45], which limits the potential performance of the Siamese architecture. In this work, we propose Deformable Siamese Attention Networks, referred as SiamAttn, to improve the feature learning capability of Siamese-based trackers. We present a new deformable Siamese attention which can improve the target representation with strong robustness to large appearance variations, and also enhance the target discriminability against distractors and complex backgrounds, resulting in more accurate and stable tracking, as shown in Fig. 1. The main contributions of this work are:

- We introduce a new Siamese attention mechanism that computes deformable self-attention and cross-attention jointly. The self-attention captures rich context information via spatial attention, and at the same time, selectively enhances interdependent channel-wise features with channel attention. The cross-attention aggregates meaningful contextual interdependencies between the target template and the search image, which are encoded into the target template adaptively to improve discriminability.
- We design a region refinement module by computing depth-wise cross correlations between the attentional features. This further enhances feature representations, leading to more accurate tracking by generating both bounding box and mask of the object.
- Our method achieves new state-of-the-art results on six benchmarks. It outperforms recent strong baselines, such as SiamRPN++ [24] and SiamMask [36], by a large margin. For example, it improves SiamRPN++ by 0.464→0.537 and 0.415→0.470 (EAO) on VOT 2016 and 2018, while keeping real-time running speed using ResNet-50 [15].

## 2. Related Work

Correlation filter based trackers have been widely used since MOSSE [3], due to their efficiency and expansibility. However, the tracking object can be continuously improved online, which inevitably limits the representation ability of such trackers. Deep learning technologies provide a powerful tool with the ability to learn strong deep representations, and recent work attempted to incorporate the correlation filter framework with such features learning capability, such as MDNet [31], C-COT [8], ECO [7] and GFS-DCF [40].

There is another trend to build trackers on Siamese networks, by learning from massive data offline. Bertinetto *et al.* [1] first introduced SiamFC for visual tracking, by using Siamese networks to measure the similarity between target and search image. Then Li *et al.* [25] applied a region proposal network (RPN) [13] into Siamese networks, referred as SiamRPN. Zhu *et al.* [44] extended the SiamRPN by developing distractor-aware training. Recently, SiamDW-RPN [43] and SiamRPN++ [24] were proposed, which allow the Siamese-based trackers to explore deeper networks, while Wang *et al.* [36] developed a SiamMask that incorporates instance segmentation into tracking. Our work is related to that of [11] where a C-RPN was developed to progressively refine the location of target with a sequence of RPNs, but we design a new module that only refines a single output region, which is particularly lightweight and can be integrated into very deep Siamese networks.

However, Siamese-based trackers can be affected by distractors with complex backgrounds. Recent work attempted to design various strategies to update the template online, in an effort to improve the target-discriminability of Siamese-based trackers, such as MLT [4], UpdateNet [42] and GradNet [26]. An alternative solution is to extend existing online discriminative framework with deep networks for end-to-end learning, e.g., ATOM [6] and DiMP [2]. In addition, Zhu *et al.* [45] exploited motion information in Siamese networks to improve the feature representation.

Recently, the attention mechanism has been widely applied in various tasks. Hu *et al.* [18] proposed a SENet to enhance the representational power of the networks by modeling channel-wise relationships via attentions. Wang *et al.* [37] developed a non-local operation in the space-time dimension to guide the aggregation of contextual information. In [12], a self-attention mechanism was introduced to harvest the contextual information for semantic segmentation. Particularly, Wang *et al.* [35] proposed a RASNet by developing an attention mechanism for Siamese trackers, but it only utilizes the template information, which might limit its representation ability. To better explore the potentials of feature attentions in Siamese networks, we compute both self-attention and cross-branch attention jointly with deformable operations to enhance the discriminative representation of target.

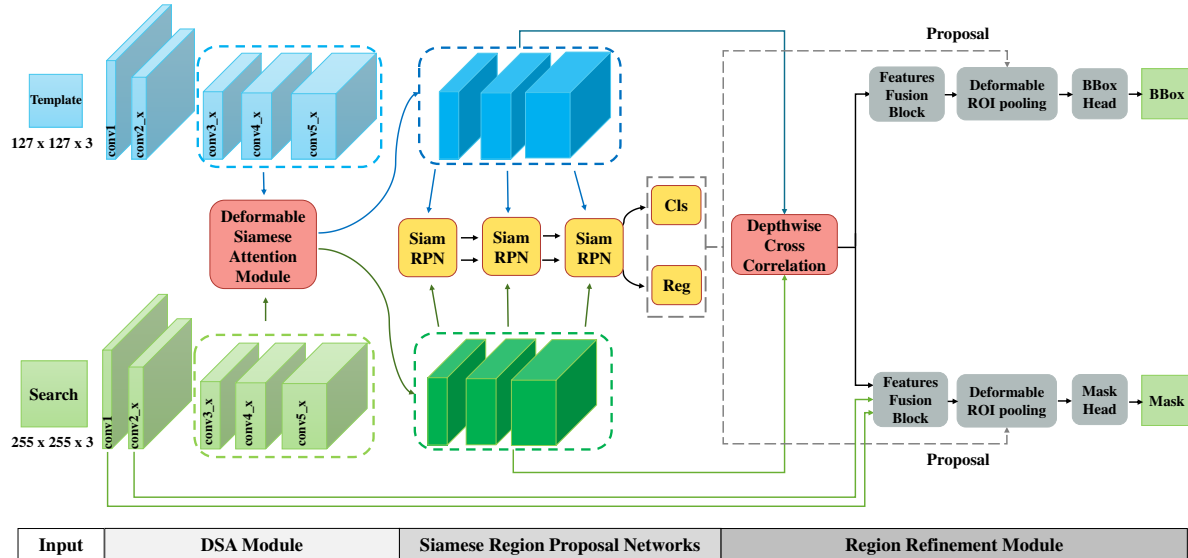


Figure 2: An overview of the proposed Deformable Siamese Attention Networks (SiamAttn). It consists of a deformable Siamese attention (DSA) module, Siamese region proposal networks (SiamRPN) and a region refinement module. The features of the last three stages are extracted and then modulated by the DSA module. It generates two-stream attentional features which are fed into SiamRPN blocks to predict a single tracking region, further refined by the refinement module.

### 3. Deformable Siamese Attention Networks

We describe the details of our Deformable Siamese Attention Networks (SiamAttn). As shown in Fig. 2, it consists of three main components: a deformable Siamese attention (DSA) module, Siamese region proposal networks (Siamese RPN), and a region refinement module.

**Overview.** We use a five-stage ResNet-50 as the backbone of Siamese networks, which computes increasingly high-level features as the layers become deeper. The features of the last three stages on both Siamese branches can be modulated and enhanced by the proposed DSA module, generating two-stream attentional features. Then we apply three Siamese RPN blocks described in [24] to the attentional features, generating dense response maps, which are further processed by a classification head and a bounding box regression head to predict a single tracking region. Finally, the generated tracking region is further refined by a region refinement module, where depth-wise cross correlations are computed on the two-stream attentional features. The correlated features are further fused and enhanced, and then are used for refining the tracking region via joint bounding box regression and target mask prediction.

#### 3.1. Siamese-based Trackers

Bertinetto *et al.* [1] introduced Siamese networks for visual object tracking, which formulates visual object tracking as a similarity learning problem. Siamese networks consist of a pair of CNN branches with sharing parameters  $\phi$ ,

which are used to project the target image ( $z$ ) and the search image ( $x$ ) into a common embedding space, where a similarity metric  $g$  can be computed to measure the similarity between them,  $g(\phi(x), \phi(z))$ . Li *et al.* [25] applied region proposal networks (RPN) [13] with Siamese networks for visual object tracking (referred as SiamRPN), where the computed features  $\phi(x)$  and  $\phi(z)$  are fed into the RPN framework using an up-channel cross-correlation operation. This generates dense response maps where RPN-based detection can be implemented, leading to significant performance improvements.

**SiamRPN++.** In [24], SiamRPN++ was introduced to improve the performance of SiamRPN, by exploring the power of deeper networks. A spatial aware sampling strategy was developed to address a key limitation of the Siamese-based trackers, allowing them to benefit from a deeper backbone like ResNet-50. Furthermore, SiamRPN++ adopts a depth-wise cross correlation to replace the up-channel cross correlation, which reduces the number of parameters and accelerates the training process. Moreover, it aggregates multi-layer features to predict the target more accurately. Similarly, we use ResNet-50 as backbone, with depth-wise cross correlation and multi-layer aggregation strategy, by following SiamRPN++ [24]. But we introduce a new Siamese attention module that enhances the learned discriminative representations of the target object and the search image, which is the key to improve the tracking performance on both accuracy and robustness.

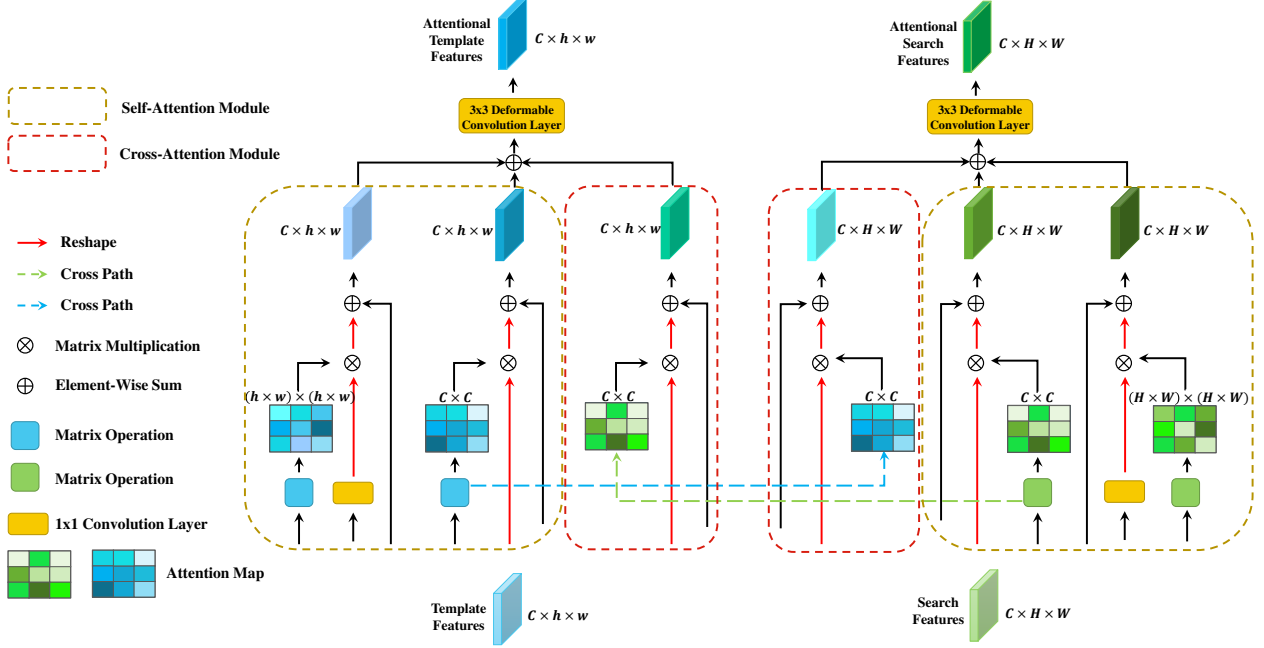


Figure 3: The proposed Deformable Siamese Attention (DSA) module, which consists of two sub-modules: self-attention sub-module and cross-attention sub-module. It takes template features and search features as inputs, and computes corresponding attentional features. The self-attention can learn strong context information via spatial attention, and at the same time, selectively emphasizes interdependent channel-wise features with channel attention. The cross-attention aggregates rich contextual interdependencies between the target template and the search image.

### 3.2. Deformable Siamese Attention Module

As illustrated in Fig. 3, the proposed DSA module takes a pair of convolutional features computed from Siamese networks as inputs, and outputs the modulated features by applying the Siamese attention mechanism. The DSA module consists of two sub-modules: a *self-attention* sub-module and a *cross-attention* sub-module. We denote the feature maps of the target and the search image as  $\mathbf{Z}$  and  $\mathbf{X}$ , with feature shapes of  $C \times h \times w$  and  $C \times H \times W$ .

**Self-Attention.** Inspired by [12], our self-attention sub-module attends to two aspects, namely channels and special positions. Unlike the classification or detection task where object classes are pre-defined, visual object tracking is a class-agnostic task and the class of object is fixed during the whole tracking process. As observed in [24], each channel map of the high-level convolutional features usually responses for a specific object class. Equally treating the features across all channels will hinder the representation ability. Similarly, as limited by receptive fields, the features computed at each spatial position of the maps can only capture the information from a local patch. Therefore, it is crucial to learn the global context from the whole image.

Specifically, self-attention is computed separately on the target branch and the search branch, and both channel self-attention and spatial self-attention are calculated at each branch. Taking the spatial self-attention for example. Sup-

pose the input features are  $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ , we first apply two separate convolution layers with  $1 \times 1$  kernels on  $\mathbf{X}$  to generate query features  $\mathbf{Q}$  and key features  $\mathbf{K}$  respectively, where  $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{C' \times H \times W}$  and  $C' = \frac{1}{8}C$  is the reduced channel number. The two features are then reshaped to  $\bar{\mathbf{Q}}, \bar{\mathbf{K}} \in \mathbb{R}^{C' \times N}$  where  $N = H \times W$ . We can generate a spatial self-attention map  $\mathbf{A}_S^S \in \mathbb{R}^{N \times N}$  via matrix multiplication and column-wise softmax operations as,

$$\mathbf{A}_S^S = \text{softmax}_{col}(\bar{\mathbf{Q}}^T \bar{\mathbf{K}}) \in \mathbb{R}^{N \times N}. \quad (1)$$

Meanwhile, a  $1 \times 1$  convolution layer with a reshape operation is applied to the features  $\mathbf{X}$  to generate value features  $\bar{\mathbf{V}} \in \mathbb{R}^{C \times N}$ , which are multiplied with the attention map and then are added to the reshaped features  $\bar{\mathbf{X}} \in \mathbb{R}^{C \times N}$  with a residual connection as,

$$\bar{\mathbf{X}}_S^S = \alpha \bar{\mathbf{V}} \mathbf{A}_S^S + \bar{\mathbf{X}} \in \mathbb{R}^{C \times N}. \quad (2)$$

where  $\alpha$  is a scalar parameter. The outputs are then reshaped back to the original size as  $\mathbf{X}_S^S \in \mathbb{R}^{C \times H \times W}$ .

We can compute channel self-attention  $\mathbf{A}_C^S$  and the channel-wise attentional features  $\mathbf{X}_C^S$  in a similar manner. Notice that on computing the channel self-attention and the corresponding attentional features, the query, key and value features are the original convolutional features computed directly from the Siamese networks, without implementing



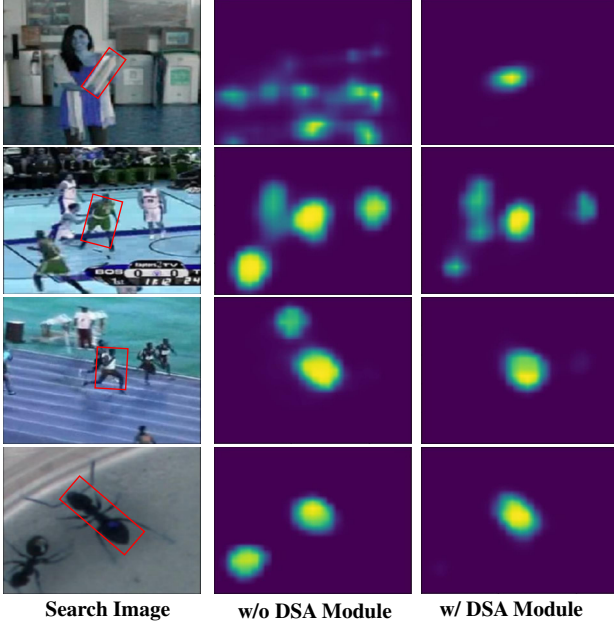


Figure 4: Visualization of confidence maps. The 1<sup>st</sup> column: search images, the 2<sup>nd</sup> column: confidence maps without our DSA module, and the 3<sup>rd</sup> column: confidence maps with DSA module which enhances target-background discriminability in the computed attentional features.

$1 \times 1$  convolutions. The final self-attentional features  $\mathbf{X}^S$  are generated by simply combining the spatial and channel-wise attentional features using element-wise sum.

**Cross-Attention.** Siamese networks usually make predictions in the last stage, while the features from two branches are computed separately, but may compensate each other. It is common that multiple objects appear at the same time, even with occlusions during tracking. Therefore, it is of great importance for the search branch to learn the target information, which enables it to generate a more discriminative representation that helps identify the target more accurately. Meanwhile, the target representation can be more meaningful when the contextual information from the search image is encoded. To this end, we propose a cross-attention sub-module to learn such mutual information from two Siamese branches, which in turn allows the two branches to work more collaboratively.

Specifically, we use  $\mathbf{Z} \in \mathbb{R}^{C \times h \times w}$  and  $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$  to denote template features and search features, respectively. Taking the search branch for example, we first reshape the target features  $\mathbf{Z}$  to  $\bar{\mathbf{Z}} \in \mathbb{R}^{C \times n}$  where  $n = h \times w$ . Then we compute the cross-attention from the target branch by performing similar operations as channel self-attention,

$$\mathbf{A}^C = \text{softmax}_{\text{row}}(\bar{\mathbf{Z}}\bar{\mathbf{Z}}^T) \in \mathbb{R}^{C \times C}. \quad (3)$$

where row-wise softmax is implemented on the computed matrix. Then the cross-attention computed from the target

branch is encoded into the search features  $\mathbf{X}$  as,

$$\bar{\mathbf{X}}^C = \gamma \mathbf{A}^C \bar{\mathbf{X}} + \bar{\mathbf{X}} \in \mathbb{R}^{C \times N}. \quad (4)$$

where  $\gamma$  is a scalar parameter, and the reshaped features  $\bar{\mathbf{X}}^C \in \mathbb{R}^{C \times H \times W}$  are the output of the sub-module.

Finally, the self-attentional features  $\mathbf{X}^S$  and the cross-attentional features  $\bar{\mathbf{X}}^C$  are simply combined with an element-wise sum, generating the attentional features for the search image. The attentional features for target image can be computed in a similar manner.

**Deformable Attention.** The building units in CNNs, such as convolution or pooling units, often have fixed geometric structures, by assuming the objects are rigid. For object tracking, it is of importance to model complex geometric transformations because the tracking objects usually have large deformations due to various factors, such as viewpoint, pose, occlusion and so on. The proposed attention mechanism can handle such challenges to some extent. We further introduce deformable attention to enhance the capability for handling such geometric transformations.

The deformable attention can sample the input feature maps at variable locations instead of the fixed ones, making them attend to the content of objects with deformations. Therefore, it is particularly suitable for object tracking, where the visual appearance of a target can be changed significantly over time. Specifically, a  $3 \times 3$  deformable convolution [5] is further applied to the computed attentional features, generating the final attentional features which are more accurate, discriminative and robust. As shown in Fig. 4, with our DSA module, the confidence maps of the attentional features focus more accurately on the interested objects, making the objects more discriminative against distractors and background.

**Region Proposals.** The DSA module outputs Siamese attentional features for both target image and search image. Then we apply three Siamese RPN blocks on the attentional features for generating a set of target proposals, with corresponding bounding boxes and class scores, as shown in Fig. 2. Specifically, a Siamese RPN block is a combination of multiple fully convolutional layers, depth-wise cross correlation, with a regression head and a classification head on top, as described in [25]. It takes a pair of convolutional features computed from the two branches of Siamese networks, and outputs dense prediction maps. By following [24], we apply three Siamese RPN blocks for the Siamese features computed from the last three stages, generating three prediction maps which are further combined by a weighted sum. Each spatial position of the combined maps predicts a set of region proposals, corresponding to the pre-defined anchors. Then the predicted proposal with the highest classification score is selected as the output tracking region.

### 3.3. Region Refinement Module

We further develop a region refinement module to improve the localization accuracy of the predicted target region. We first apply depth-wise cross correlations between the two attentional features across multiple stages, generating multiple correlation maps. Then the correlation maps are fed into a fusion block, where the feature maps with different sizes are aligned in both spatial and channel domains, e.g., by using up-sampling or down-sampling, with  $1 \times 1$  convolution. Then the aligned features are further fused (with element-wise sum) for predicting both bounding box and mask of the target. Besides, we further perform two additional operations: (1) we combine the convolutional features of the first two-stages into the fused features, which encodes richer local detailed information for mask prediction; (2) a deformable RoI pooling [5] is applied to compute target features more accurately. Bounding box regression and mask prediction often require different levels of convolutional features. Thus we generate the convolutional features with spatial resolutions of  $64 \times 64$  for mask prediction and  $25 \times 25$  for bounding box regression.

Notice that the classification head is not applied since visual object tracking is a class-agnostic task. The input resolution for the bounding box head is  $4 \times 4$ . By using two fully-connected layers with 512 dimensions, the bounding box head predicts a 4-tuple  $t = (t_x, t_y, t_w, t_h)$ . Similarly, the input of mask prediction head has a spatial resolution of  $16 \times 16$ . By using four convolutional layers and a de-convolutional layer, the mask head predicts a class-agnostic binary mask with  $64 \times 64$  for the tracking object. Compared to ATOM [6] and SiamMask [36] which predict bounding boxes and masks *densely*, our refinement module uses light-weight convolutional heads to predict a bounding box and a mask for a single tracking region, which is much more computationally efficient.

### 3.4. Training Loss

Our model is trained in an end-to-end fashion, where the training loss is a weighted combination of multiple functions from Siamese RPN and region refinement module:

$$L = L_{\text{rpn-cls}} + \lambda_1 L_{\text{rpn-reg}} + \lambda_2 L_{\text{refine-box}} + \lambda_3 L_{\text{refine-mask}}. \quad (5)$$

where  $L_{\text{rpn-cls}}$  and  $L_{\text{rpn-reg}}$  refer to a classification loss and a regression loss in Siamese RPN. We employ a negative log-likelihood loss and a smooth L1 loss correspondingly. Similarly,  $L_{\text{refine-box}}$  and  $L_{\text{refine-mask}}$  indicate a smooth L1 loss for bounding box regression and a binary cross-entropy loss for mask segmentation in region refinement module. The weight parameters  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are used to balance different tasks, which are empirically set to 0.2, 0.2 and 0.1 in our implementation.

## 4. Experiments and Results

We conduct extensive experiments on six benchmark databases: OTB-2015 [38], UAV123 [29], VOT2016 [21], VOT2018 [22], LaSOT [10] and TrackingNet [30] datasets and provide ablation study to verify the effects of each proposed component.

### 4.1. Datasets

**OTB-2015 [38].** OTB-2015 is one of the most commonly used benchmarks for visual object tracking. It has 100 fully annotated video sequences using two evaluation metrics, a precision score and an area under curve (AUC) of success plot. The precision score is the percentage of frames in which the distance between the center of tracking results and ground-truth is under 20 pixels. The success plot shows the ratios of successfully tracked frames with various thresholds.

**VOT2016 [21] & VOT2018 [22].** VOT2016 and VOT2018 are widely-used benchmarks for visual object tracking. VOT2016 contains 60 sequences with various challenging factors while VOT2018 has 10 different sequences with VOT2016. The two datasets are annotated with the rotated bounding boxes, and a reset-based methodology is applied for evaluation. For both benchmarks, trackers are measured in terms of accuracy (A), robustness (R), and expected average overlap (EAO).

**UAV123 [29].** UAV123 contains 123 sequences captured from low-altitude UAVs. Unlike other tracking datasets, the viewpoint of UAV123 is aerial and the targets to be tracked are usually small.

**LaSOT [10].** LaSOT is a large-scale dataset with 1400 sequences in total, and 280 sequences in test set. high-quality dense annotations are provided, and deformation and occlusion are very common in LaSOT. The average sequence length of LaSOT is 2500 frames, demonstrating long-term performance of the evaluated trackers.

**TrackingNet [30].** TrackingNet contains 30000 sequences with 14 million dense annotations and 511 sequences in the test set. It covers diverse object classes and scenes, requiring trackers to have both discriminative and generative capabilities.

### 4.2. Implementation Details

We use ResNet-50, pre-trained on ImageNet [9], as the backbone, and the whole networks are then fine-tuned on the training sets of COCO [27], YouTube-VOS [39], LaSOT [10] and TrackingNet [30]. We apply stochastic gradient descent (SGD) with a momentum of 0.9 and a weight decay of  $10^{-5}$  for optimization. By following SiamFC [1], we adopt an exemplar image of  $127 \times 127$  and a search image of  $255 \times 255$  for training and testing.

Our model is trained for 20 epochs. By following SiamRPN++ [24], we use a warm-up learning rate of  $10^{-3}$

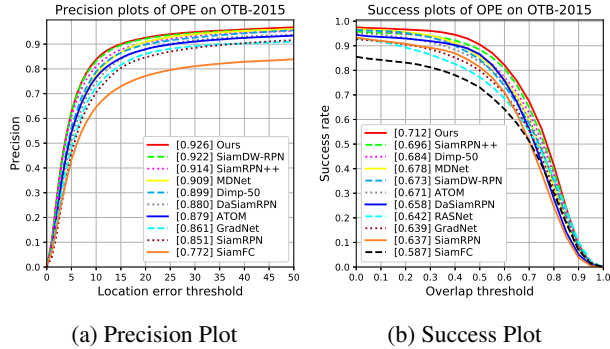


Figure 5: Comparisons with state-of-the-art methods on success and precision plots on OTB-2015 dataset.

for the first 5 epochs which decays exponentially from  $5 \times 10^{-3}$  to  $5 \times 10^{-4}$  for the last 15 epochs. The weights of backbone are frozen for the first 10 epochs, then the whole networks are trained end-to-end for the last 10 epochs. In particular, the learning rate of backbone is 20 times smaller than the other parts. Batch size is set to 12. Our anchor boxes have 5 aspect ratios,  $[0.33, 0.5, 1, 2, 3]$ . In Siamese-RPN blocks, an anchor box is labelled as positive when it has an  $\text{IoU} > 0.6$ , or as negative when  $\text{IoU} < 0.3$ . Other patches whose  $\text{IoU}$  overlap falls in between are ignored. In addition, we sample 16 regions from each image with  $\text{IoU} > 0.5$  for training our region refinement module.

For the backbone networks, we employ dilated convolutions for the last two blocks to increase the receptive fields, and the effective strides at these two blocks are reduced from 16 or 32 pixels to 8 pixels. We also reduce the number of feature channels to 256 for the last three blocks of the backbone networks via a  $1 \times 1$  convolution layer. During inference, cosine window penalty, scale change penalty and linear interpolation update strategy [25] are applied. Only one single region with the highest score predicted by Siamese RPN blocks is fed into our region refinement module. Our method is implemented using PyTorch, and we use NVIDIA GeForce RTX 2080Ti GPU.

### 4.3. Comparisons with the State-of-the-art

**On OTB-2015.** Fig. 5 shows quantitative results on OTB-2015 dataset. Our tracker achieves the best AUC and precision score among all methods for this widely studied dataset. Specifically, we obtain a precision of 0.712 and an AUC of 0.926 which surpass that of SiamRPN++ [24] by 1.6% and 1.2%, respectively.

**On VOT2016 & VOT2018.** The results on VOT2016 and VOT2018 are compared in Tab. 1. Our tracker achieves 0.68 accuracy, 0.14 robustness and 0.537 EAO on VOT2016, outperforming the state-of-the-art methods under all metrics. Compared with recent SiamRPN++ [24] and SiamMask [36], our method has significant improvements of 7.3% and 9.5% on EAO respectively. On

Tracker	VOT2016			VOT2018		
	A $\uparrow$	R $\downarrow$	EAO $\uparrow$	A $\uparrow$	R $\downarrow$	EAO $\uparrow$
SiamFC [1]	0.53	0.46	0.235	0.50	0.59	0.188
MDNet [31]	0.54	0.34	0.257	-	-	-
C-COT [8]	0.54	0.24	0.331	0.49	0.32	0.267
FlowTrack [45]	0.58	0.24	0.334	-	-	-
SiamRPN [25]	0.56	0.26	0.344	-	-	-
C-RPN [11]	0.59	-	0.363	-	-	-
ECO [7]	0.55	0.20	0.375	0.48	0.28	0.276
DaSiamRPN [44]	0.61	0.22	0.411	0.59	0.28	0.383
SPM [34]	0.62	0.21	0.434	-	-	-
SiamMask-Opt [36]	0.67	0.23	0.442	0.64	0.30	0.387
UpdateNet [42]	0.61	0.21	0.481	-	-	0.393
GFS-DCF [40]	-	-	-	0.51	0.14	0.397
ATOM [6]	-	-	-	0.59	0.20	0.401
SiamRPN++ [24]	0.64	0.20	0.464	0.60	0.23	0.415
Dimp-50 [2]	-	-	-	0.60	0.15	0.440
Ours	0.68	0.14	0.537	0.63	0.16	0.470

Table 1: Results on VOT2016 and VOT2018, with accuracy (A), robustness (R) and expected average overlap (EAO).

	ARCF [19]	ECO [7]	SiamRPN [25]	DaSiam-RPN [44]	SiamRPN++ [24]	ATOM [6]	Dimp-50 [2]	Ours
AUC	0.47	0.525	0.527	0.586	0.613	0.644	0.654	0.650
Pr	0.67	0.741	0.748	0.796	0.807	-	-	0.845

Table 2: Results on UAV123.

VOT2018, our method achieves the top EAO score while having competitive accuracy and robustness with other state-of-the-art methods. SiamMask-Opt [36] attains the best accuracy by finding the optimal rotated rectangle from a binary mask, which however increases the computational cost significantly, and reduces its fps to 5. Our method only uses a single rotated minimum bounding rectangle from the predicted mask, which achieves a comparable accuracy of 0.63, but has a large improvement on EAO,  $0.387 \rightarrow 0.470$ , with a real-time running speed at 33 fps. Compared with SiamRPN++ and recent leading tracker Dimp-50 [2], our tracker obtains a clear performance gain of 5.5% and 3.0% respectively in terms of EAO, demonstrating the efficiency of the proposed Siamese attention and refinement module.

**On UAV123.** As shown in Tab. 2, SiamAttn attains the best precision, improving the closest one: SiamRPN++, from 0.807 to 0.845, while having a comparable AUC with DiMP-50 which did not report the precision score.

**On LaSOT.** Tab. 3 shows the comparison results on LaSOT with long sequences. Our method attains the best normalized precision, outperforming SiamRPN++ considerably by 56.9%  $\rightarrow$  64.8% (with 49.5%  $\rightarrow$  56.0% on success). Again, our method has a comparable success score with DiMP-50, while attaining a higher normalized precision.

**On TrackingNet.** We further evaluate SiamAttn on the large-scale TrackingNet. As illustrated in Tab. 4, it outper-

	MLT [4]	MDNet [31]	DaSiam-RPN [44]	Update-Net [42]	SiamRPN++ [24]	ATOM [6]	Dimp-50 [2]	Ours
Success(%)	34.5	39.7	41.5	47.5	49.5	51.5	56.9	56.0
Norm.Pr(%)	-	46.0	49.6	56.0	56.9	57.6	64.3	64.8

Table 3: Results on LaSOT.

	GFS-DCF [40]	DaSiam-RPN [44]	Update-Net [42]	ATOM [6]	SPM [34]	SiamRPN++ [24]	Dimp-50 [2]	Ours
Success(%)	60.9	63.8	67.7	70.3	71.2	73.3	74.0	75.2
Norm.Pr(%)	71.2	73.3	75.2	77.1	77.8	80.0	80.1	81.7

Table 4: Results on TrackingNet.

Method	A $\uparrow$	R $\downarrow$	EAO $\uparrow$	$\Delta$ EAO
Baseline	0.64	0.20	0.464	-
Baseline+ML	0.66	0.21	0.477	+1.3%
Baseline+RR	0.67	0.19	0.486	+2.2%
Baseline+RR+SA	0.66	0.16	0.511	+4.7%
Baseline+RR+CA	0.67	0.15	0.513	+4.9%
Baseline+RR+CA+SA (Ours)	0.68	0.14	0.537	+7.3%

Table 5: Ablation study on VOT2016. SiamRPN++ is baseline. ML: Mask Learning, RR: Region Refinement (including ML), SA: Self-Attention, and CA: Cross-Attention.

forms all previous methods consistently. Compared with the most recent DiMP-50, SiamAttn has improvements of 1.2% on success, and 1.6% on normalized precision, demonstrating its ability to handle diverse objects over complex scenes.

#### 4.4. Ablation Study

We study the impact of individual components in SiamAttn, and conduct ablation study on VOT2016.

**Model Architecture.** We use SiamRPN++ [24] as baseline. As shown in Tab. 5, SiamRPN++ achieves an EAO of 0.464. By adding mask learning layers to SiamRPN++, the EAO can be improved to 0.477. With our region refinement module, the EAO score is further increased by +2.2%. Compared with the baseline, the accuracy score improves from 0.64 to 0.67, with comparable robustness. Our Siamese attention consists of both self-attention and cross-attention, each of which can further improve the EAO by +4.7% or +4.9% respectively. This suggests that the proposed cross-attention is critical to the tracking results, and even has a more significant impact than the self-attention. Jointly exploring both self-attention and cross-attention makes our method not only robust, but also more accurate. This results in a high EAO of 0.537, surpassing the baseline by a large margin of 7.3%.

**Deformable Layers.** In this study, we evaluate the impact of deformable operations by replacing them with a regular convolution. As shown in Tab. 6, this results in slight performance drops, e.g., 0.537 $\rightarrow$ 0.520 EAO with deformable convolution and 0.537 $\rightarrow$ 0.531 EAO with

Deformable convolution	Deformable Pooling	A $\uparrow$	R $\downarrow$	EAO $\uparrow$
$\times$	$\times$	0.67	0.15	0.516
$\times$	$\checkmark$	0.67	0.15	0.520
$\checkmark$	$\times$	0.68	0.15	0.531
$\checkmark$	$\checkmark$	0.68	0.14	0.537

Table 6: The impact of deformable layers on VOT2016.

Method	Training set	A $\uparrow$	R $\downarrow$	EAO $\uparrow$
SiamAttn	VID, YTB-BB, COCO, DET, YTB-VOS	0.68	0.15	0.525
SiamAttn	COCO, YTB-VOS, LaSOT, TrackingNet	0.68	0.14	0.537

Table 7: Results on VOT2016 with training sets as listed.

deformable pooling. By removing all deformable layers, our model can still achieve an EAO of 0.516, compared favorably against SiamRPN++ with 0.464, suggesting that the proposed Siamese attention and refinement modules are the primary contributors to the performance boost.

**On Training Data.** In this study, we investigate the impact of training with different training sets. Our current results are achieved by using a combination of multiple training sets from recent LaSOT [10] and TrackingNet [30], with COCO [27] and YouTube-VOS [39], mainly following [6] with an additional YouTube-VOS [39] for providing mask annotations. We also report the results on a different training combination applied by [36], including COCO [27], YouTube-VOS [39], YouTube-BoundingBox [32], ImageNet-VID, and ImageNet-Det [33]. Results are reported in Tab. 7. Using the recent large-scale tracking datasets can improve the results with 1.2% EAO on VOT2016, while our approach can still achieve the state-of-the-art performance using a different choice of the training sets.

**Speed Analysis.** On OTB-2015, UAV, LaSOT and TrackingNet benchmarks, our model predicts axis-aligned bounding box, without mask head. It can achieve an inference speed of 45 fps. On VOT benchmarks, our model generates the rotated boxes from the predicted masks, which reduces the inference speed to 33 fps.

## 5. Conclusion

We have presented new Deformable Siamese Attention Networks for visual object tracking. We introduce a deformable Siamese attention mechanism consisting of both self-attention and cross-attention. The new Siamese attention can strongly enhance target discriminability, and at the same time, improve the robustness against large appearance variations, complex backgrounds and distractors. Additionally, a region refinement module is designed to further increase the tracking accuracy. Extensive experiments are conducted on six benchmarks, where our method obtains new state-of-the-art results, with real-time running speed.



## References

- [1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016.
- [2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019.
- [3] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.
- [4] Janghoon Choi, Junseok Kwon, and Kyoung Mu Lee. Deep meta learning for real-time target-aware visual tracking. In *ICCV*, 2019.
- [5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.
- [6] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019.
- [7] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017.
- [8] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [10] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019.
- [11] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019.
- [12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019.
- [13] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.
- [14] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, 2017.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [16] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016.
- [17] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014.
- [18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.
- [19] Ziyuan Huang, Changhong Fu, Yiming Li, Fuling Lin, and Peng Lu. Learning aberrance repressed correlation filters for real-time uav tracking. In *ICCV*, 2019.
- [20] Zdenek Kalal, Krystian Mikołajczyk, and Jiri Matas. Tracking-learning-detection. *TPAMI*, 34(7):1409–1422, 2011.
- [21] M Kristan, A Leonardis, J Matas, M Felsberg, R Pflugfelder, L Čehovin, T Vojř, G Häger, A Lukežič, G Fernández, et al. The visual object tracking vot2016 challenge results. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9914:777–823, 2016.
- [22] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojř, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *ECCV*, 2018.
- [23] Kuan-Hui Lee and Jenq-Neng Hwang. On-road pedestrian tracking across multiple driving recorders. *IEEE Transactions on Multimedia*, 17(9):1429–1438, 2015.
- [24] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.
- [25] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.
- [26] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Gradnet: Gradient-guided network for visual object tracking. In *ICCV*, 2019.
- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- [28] Liwei Liu, Junliang Xing, Haizhou Ai, and Xiang Ruan. Hand posture recognition using finger geometric feature. In *ICPR*, 2012.
- [29] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016.
- [30] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.
- [31] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.
- [32] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *CVPR*, 2017.
- [33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.
- [34] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Spm-tracker: Series-parallel matching for real-time visual object tracking. In *CVPR*, 2019.
- [35] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *CVPR*, 2018.

- [36] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.
- [37] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
- [38] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013.
- [39] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.
- [40] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Joint group feature selection and discriminative filter learning for robust visual object tracking. In *ICCV*, 2019.
- [41] Tianyu Yang and Antoni B. Chan. Learning dynamic memory networks for object tracking. In *ECCV*, 2018.
- [42] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, 2019.
- [43] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 2019.
- [44] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018.
- [45] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *CVPR*, 2018.