Neural Processing Letters (2024) 56:98
https://doi.org/10.1007/s11063-024-11556-6

SiamRAAN: Siamese Residual Attentional Aggregation Network for Visual Object Tracking

Zhiyi Xin[1] · Junyang Yu[1] · Xin He[1] · Yalin Song[1] · Han Li[1]

Accepted: 11 February 2024

Abstract
The Siamese network-based tracker calculates object templates and search images independently, and the template features are not updated online when performing object tracking. Adapting to interference scenarios with performance-guaranteed tracking accuracy when background clutter, illumination variation or partial occlusion occurs in the search area is a challenging task. To effectively address the issue with the abovementioned interference and to improve location accuracy, this paper devises a Siamese residual attentional aggregation network framework for self-adaptive feature implicit updating. First, SiamRAAN introduces Self-RAAN into the backbone network by applying residual self-attention to extract effective objective features. Then, we introduce Cross-RAAN to update the template features online by focusing on the high-relevance parts in the feature extraction process of both the object template and search image. Finally, a multilevel feature fusion module is introduced to fuse the RAAN-enhanced feature information and improve the network's ability to perceive key features. Extensive experiments conducted on benchmark datasets (GOT-10K, LaSOT, OTB-50, OTB-100 and UAV123) demonstrated that our SiamRAAN delivers excellent performance and runs at 51 FPS in various challenging object tracking tasks. Code is available at https://github.com/MallowYi/SiamRAAN.

Keywords Object tracking · Siamese network · Attentional aggregation network · Multilevel feature fusion

✉ Junyang Yu
jyyu@henu.edu.cn

Zhiyi Xin
henuxzy612@henu.edu.cn

Xin He
hxsyjkf@foxmail.com

Yalin Song
yalinsong@outlook.com

Han Li
lihan@henu.edu.cn

[1] School of Software, Henan University, Kaifeng 475000, China

Published online: 11 March 2024

1 Introduction

Visual object tracking is one of the critical research topics in the field of computer vision [1, 2]. It serves as a fundamental task in computer vision and is extensively applied in the fields of public security, human-computer interaction and autonomous driving. Visual object tracking algorithms are required to give an object's initial position and size in a video sequence and to achieve a continuous and stable tracking of the object in subsequent frames. Despite many advances in the field recently [3–5], it remains a challenging task to achieve long-term stable object tracking.

Recently, deep learning has evolved rapidly and many advances have been made in feature learning [6–9], and it has demonstrated impressive performance in the field of computer vision [10, 11]. Consequently, deep learning-based object-tracking algorithms have been introduced successively. Among them, Siamese networks have attracted more attention and research due to their higher computational speed compared to other deep learning algorithmic frameworks. For example, Bertinetto et al [12] introduced the Siamese network for visual object tracking, which first transformed the visual object tracking task into an object matching problem by learning a generic similarity mapping through the inter-correlation operation between the object template and the search region. Liu et al [13] proposed a multi-level similarity model to improve the tracker's recognition of semantic interference. SiamRPN [14] introduced a region proposal network that leverages classification and regression branching to distinguish object-background regions and fine-tune candidate regions. Recent work such as DaSiamRPN [15], SiamRPN++ [16], and C-RPN [17] have enhanced SiamRPN. SiamCAR [3] devised an anchor-free and proposal-free framework and decomposed the tracking problem into two sub-problems, pixel classification and regression at that pixel, to solve the object tracking problem in a pixel-by-pixel manner.

Siamese-based trackers are trained completely offline using a large number of frame pairs collected from video, so there exists the problem that object templates can not be updated online, it exposed inevitably potential risk for tracking drift, especially for these accurate tracking objects with highly variable appearance, similarities or occlusion. Furthermore, in the Siamese architecture, the characteristics of the target object and the search image are calculated independently, where background context information is completely discarded in the target feature, but background information is important for distinguishing between targets and interferers. Recent work [18, 19] attempted to enhance the object representation by integrating features of the preorder object with neglect of the distinguished context information in the background. In [20, 21], the attention and depth characteristics of the target template and the search image are calculated separately, while the template features remain unchanged during tracking, which limits the potential performance of the Siamese architecture.

In this paper, we introduce a novel Siamese attention mechanism by introducing self-attention and cross-attention in Siamese network to encode the rich background contextual information into object representation, the attention mechanism enables to enhance object representation ability with strong properties against changes in appearance and to strengthen the distinguished ability for object against disruptors and complicated background information with promising outcomes of more stable and accurate tracking. We leverage the SiamCAR [3] as the benchmarks tracker and introduce Siamese Residual Attentional Aggregation Network to enhance the features learning ability of Siamese-based tracker called SiamRAAN. Meanwhile, SiamCAR utilizes compressed different depth features for correlation operations to obtain response maps, and compression causes loss of accuracy. This paper proposes MFF module to alleviate the issue, which uses completely different depth

features for the correlation operation and then fuses the different response maps to boost the accuracy of the response maps.

The main contributions of this work are as follows:

- We design a novel Siamese attention mechanism, which embeds Self-RAAN and Cross-RAAN in backbone, aiming to achieve effective mining of object features and empower the feature representation.
- We devise a multi-level feature fusion module to achieve more accurate tracking by calculating the depth cross-correlation between different feature layers and then fusing and compressing multiple response maps, effectively enhancing the inter-correlation response maps of shallow features.
- Extensive evaluations on the GOT-10K, LaSOT, OTB-50, OTB-100 and UAV123 datasets have demonstrated the effectiveness of proposed SiamRAAN network, particularly in successfully tracking objects in the presence of interference such as background clutter, illumination variation and partial occlusion, while tracking at a speed of 51 FPS.

This paper is organized as follows. Section 2 provides a review of the related work for the proposed method. Section 3 introduces the proposed SiamRAAN tracker. Section 4 presents the experimental results of the proposed method. Section 5 offers the conclusions.

2 Related Work

This section presents related work in two ways. Section 2.1 describes recent trackers based on Siamese networks. Section 2.2 illustrates the attention-based mechanism tracker.

2.1 Siamese-Based Tracker

Twin networks were originally applied to video object tracking tasks in SINT [22] and SiameseFC [12] published in 2016, where the algorithms first transformed the visual object tracking task into a object matching problem, learning generic similarity mappings through mutual correlation operations between object templates and search regions. The SiamRPN [14] algorithm built on SiameseFC, incorporated the idea of a region candidate network to extract target candidate boxes, and improved the accuracy of the algorithm by introducing RPN [23] to fine-tune the prediction edges of the targets. DaSiamRPN [15] improves the generalization ability of the model by training set data augmentation and also improves the discriminative ability of the model by introducing negative samples with different degrees of difficulty for training. c-RPN [17] solves the problem of data imbalance by adding cascaded rpn between different layers in the Siamese network. SiamRPN++ [16] enhances the SiamRPN in the sample sampling strategy to prevent the problem that positive samples are all located in the center of the image and affect the target localization. SiamDW [24] utilized a deeper and wider convolutional god-general network in the Siamese tracking algorithm to improve the robustness and accuracy of the algorithm. SiamCAR [3] performed classification and regression on a pixel-by-pixel basis, greatly reducing the number of hyperparameters through an anchor-free approach, allowing the tracker to be used without complex parameter tuning. SiamGAT [25] established the correspondence between the target and the search area through a graph-notice network, adapting to different object size and aspect ratio variations and improving the tracking accuracy. SiamGLM [26] proposed a Siamese network object

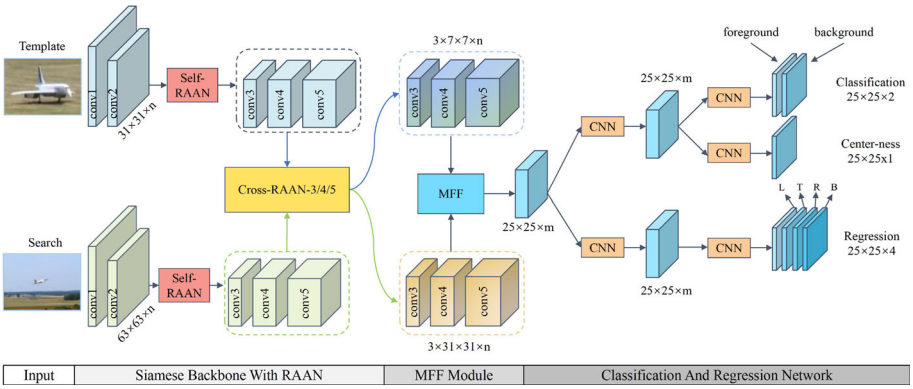

Fig. 1 The framework of the proposed Siamese Residual Attentional Aggregation Network (SiamRAAN). It consists of a Siamese backbone with residual attentional aggregation network(RAAN), a multi-level feature fusion(MFF) module and a classification and regression network

tracking algorithm based on global and local feature matching, which makes the network more robust to deformed objects.

2.2 Attention-Based Tracker

RASNet [20] introduced the attention mechanism model into the Siamese network object tracking problem and proposed three models, the generic attention mechanism, the residual attention mechanism, and the channel attention mechanism. CGACD [27] proposed a pixel-guided spatial attention module and a channel-guided channel attention module to highlight corner areas for corner detection. SiameAttn [28] suggested Deformable Siamese Attention Networks to improve the feature learning capability of the Siamese network tracker. The literature [29] provided a convolutional attention module to enhance the feature space location and the weight of the feature channels. DSN [30] designed a joint module of modal channel attention to assign weights in the feature extraction stage and improve the accuracy of tracking. In these works [21, 31], residual channel attention was introduced to determine the channel weights of the object template features using the relationship between feature channels, so that the attention of template feature extraction is focused on the channel features of the target foreground and more effective features are selected.

3 Methodology

This section elaborates the proposed SiamRAAN framework. As presented in Fig. 1, SiamRAAN includes three components: Siamese backbone with residual attentional aggregation network (RAAN) is responsible for features extraction, Self-RAAN improves the feature representation and Cross-RAAN enhances the recognition of tracked target features through information interactions, multi-level feature fusion (MFF) module coordinates to compute and fuse the inter-correlation response maps of the template with the features of the search image at different depths, and classification and regression network is designed to bounding box prediction.

3.1 Overview of SiamRAAN Framework

The proposed tracker's Siamese backbone network was constructed using the same modified five-stage ResNet-50 as SiamRPN++ [16], with a progressively larger chunk of features computed as the number of layers deepened. The network consists of two branches of template and search, the former regards the template patch Z as the input and the latter leverages search zone X as the input with the application of backbone network to effectively extract corresponding image features, meanwhile, two branches and their backbone network share the same unfilled convolutional architecture. We introduce residual attentional aggregation network (RAAN) in Siamese backbone (see in 3.2) in order to efficiently mine the relevance features between template patch and search image to improve location accuracy. Each stage of output feature stems from the template and search branch of modified five-stage ResNet-50 is defined as $\varphi_i(Z)$ and $\varphi_i(X)$, $i = \{1, 2, 3, 4, 5\}$, where the output of the second stage is fed into the subsequent stages by Self-RAAN self-enhancement, and the features of the third, fourth and fifth stages are cross-enhanced by the corresponding Cross-RAAN to generate the final attention features. The output features of template branch and search branch with the Cross-RAAN are denoted by $\psi_3(Z)$, $\psi_4(Z)$, $\psi_5(Z)$ and $\psi_3(X)$, $\psi_4(X)$, $\psi_5(X)$, respectively.

$$\begin{aligned} \varphi_3(Z) &= F_{Self-RAAN}(\varphi_2(Z)) \\ \varphi_3(X) &= F_{Self-RAAN}(\varphi_2(X)) \end{aligned} \tag{1}$$

$$\psi_i(Z), \psi_i(X) = F^i_{Cross-RAAN}(\varphi_i(Z), \varphi_i(X)), i = \{3, 4, 5\} \tag{2}$$

$F_{Self-RAAN}(\cdot)$ and $F_{Cross-RAAN}(\cdot)$ are applied to residual attention aggregation network for feature enhancement.

With the intention of improving the accuracy of identifying object's locations and their bounding boxes based on using two branches to produce comprehensive information, our model generated multiple cross-correlation response maps of different depths of the relevant layers with features extracted from the last three convolution blocks in the Siamese backbone. Specifically, the model performs the following operations:

$$R = F_{MFF}(\psi_i(Z), \psi_i(X)), i = \{3, 4, 5\} \tag{3}$$

$F_{MFF}(\cdot)$ is adopted to a multi-layer feature fusion module for fusing features of different depth-related layers.

The MFF module performs a deep correlation operation on features of different depths in two branches (Conv3, Conv4 and Conv5) to obtain response maps containing 512, 1024 and 2048 channels respectively, and then fuses these three response maps to obtain the comprehensive response map. To reduce the number of features and speed up the computation, our model leverages a 1×1 convolution kernel to dimensionality reduction calculation for the comprehensive response map, then reduce the channel dimensionality to 256, which is regarded as the input to the regression classification network.

Our module will obtain a six-dimensions vector $T_{(i,j)} = (cls, cen, l, t, r, b)$ by performing classification and regression to comprehensive response map R, where cls, cen, $l + r$ and $t + b$ denotes the prospective probability of classification at the location, the centrality score for this location, width and height in the current frame, respectively. Then the object's location and bounding box information of current frame is obtained with $T_{(i,j)}$.

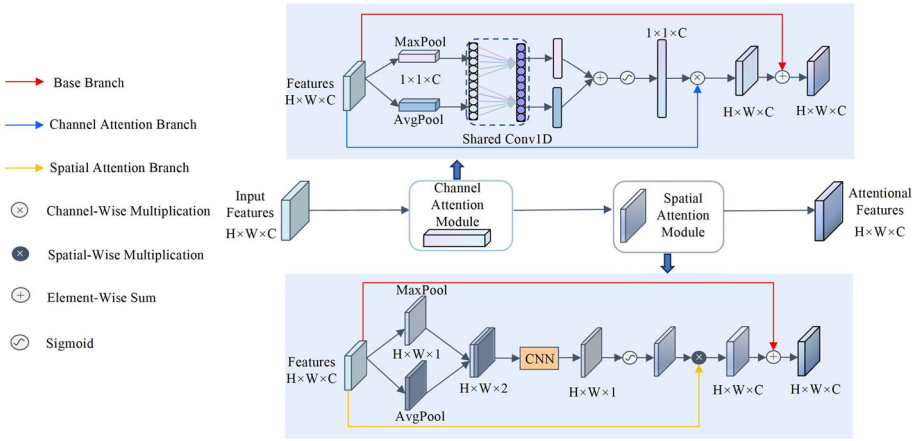

Fig. 2 The main architecture of Self-RAAN. The network contains two-order modules of channel and spatial. Complement channel-level and spatial-level feature, where the base branch retains original features, channel attention branch and spatial attention branch produces channel feature and spatial feature in response to different channel weights and channel weights respectively

3.2 Self-Residual Attentional Aggregation Network

To improve the expression ability of feature maps and tracking performance, Self-RAAN introduces attention mechanism in Siamese backbone, as shown in Fig. 1. Motivated by CBAM [32], our Self-RAAN focus on both channels and special positions, using both max-pooling and average-pooling to collect important information about object features. In contrast to tasks such as classification or detection of predefined object classes, visual object tracking is a task that is not tied to the object class and the class of the object is fixed throughout the tracking process. Each channel of advanced convolutional features maps a response that is typical to a specific object class, and handling features equally in all channels would hinder the representational power of the feature map. Also due to perceptual field limitations, features at each spatial location can only represent local information about the image. It is therefore extremely critical to learn global contextual information from the whole image.

By combining the two aspects of channels and special positions, our Self-RAAN is able to learn the category information of the tracked target, focusing on the feature representations of specific channels, and at the same time learn the context information of the spatial position, obtaining the global context information from the local features, which can better understand and differentiate the differences between the target and the background, and improve the performance and robustness of target tracking.

Specifically, the proposed Self-RAAN contains two modules channels attention and spatial attention. It can be seen from Fig. 1, with the Siamese features extraction network, template patch Z in the second stage of template branch input is transformed template feature with $31\times31\times n$, search region X in search branch input is transformed search feature with $63\times63\times n$, then template feature and search feature in their each Siamese branch is inputted into shared parameters of Self-RAAN. As presented in Fig. 2, input features of Self-RAAN finish self-adaptive enhancement operation by channel attention model and spatial attention model in order. Feature map dimensions is denoted by $\mathbb{R}^{H\times W\times C}$, where H, W and C presents the feature map of high, width and the number of channels, respectively.

In channel attention model, the model firstly performs max-pooling and average-pooling operations to preserve the channel information and compress the height and width of the feature map to 1×1 to aggregate the spatial information of the features, generating two different spatial contexts of information. Be different from CBAM [32], two pooled features were inputted into a one-dimension convolution of shared parameters rather than shared multi-level perceptron (MLP) network. While high accuracy is achieved with the help of MLP networks but without account of higher model complexity and significant computational overhead. Simultaneously, the methodology of applying dimensionality reduction before computing attention using MLP networks brings side effects to channel attention prediction, capturing the dependencies between all channels is inefficient and unnecessary. We use a one-dimensional convolution operation to capture correlations between channels more effectively with the objectives of reducing the number of model parameters and improving computation efficiency while maintaining a certain level of model expressiveness. Behind the shared Conv1D network, our network can obtain the channel attention map $A_c \in \mathbb{R}^{1\times1\times C}$ by using elements to sum and sigmoid function in accordance to max-pooling feature map $A_{cm} \in \mathbb{R}^{1\times1\times C}$ and average-pooling feature map $A_{ca} \in \mathbb{R}^{1\times1\times C}$. Eventually, channel attentional features can be obtained after multiplying the input features $X \in \mathbb{R}^{H\times W\times C}$ with the channels corresponding to the channel attention weights A_c. The process of channel feature enhancement is computed below:

$$A_{cm} = f^5(MaxPool(X)) \tag{4}$$

$$A_{ca} = f^5(AvgPool(X)) \tag{5}$$

$$A_c = \sigma\{A_{cm} + A_{ca}\} \tag{6}$$

$$M_c = A_c \times X + X \tag{7}$$

σ presents sigmoid operation and f^5 expresses the one-dimensional convolution operation with convolutional kernel size is equal to 5.

In spatial attention model, it adopts the same idea of channel attention model to aggregate information with max-pooling and average-pooling operations. For the spatial level, we first aggregate the channel information of the features by compressing the channel number C of the feature map to 1 in the same two pooling ways with the channel attention model, then stitch them together along the channel direction and utilize a standard convolution layer to perform convolution to complete the fusion of the two pooling information, and finally aid the sigmoid function to complete the computation of the spatial attention map $A_s \in \mathbb{R}^{H\times W\times1}$. Given the input features as $X \in \mathbb{R}^{H\times W\times C}$, the output of spatial attention features as $M_s \in \mathbb{R}^{H\times W\times C}$, the process of spatial feature enhancement is computed as follows:

$$X_s = Cat(MaxPool(X), AvgPool(X)) \tag{8}$$

$$A_s = \sigma\{f^{3\times3}(X_s)\} \tag{9}$$

$$M_s = A_s \times X + X \tag{10}$$

σ presents sigmoid operation and $f^{3\times3}$ exhibits the two-dimensional convolution operation with convolutional kernel size is equal to 3×3, Cat shows the splicing operation along the channel dimension.

We adopt residual architecture in channel attention model and spatial attention model and divide into base branch and attention branch. To obtain the final attention feature by summing the features of base branch and attention branch after deriving from attention features. Since the attention feature map generated by the attention branch enhances the features of the target, while the base branch retains all the features of the original image, the introduction

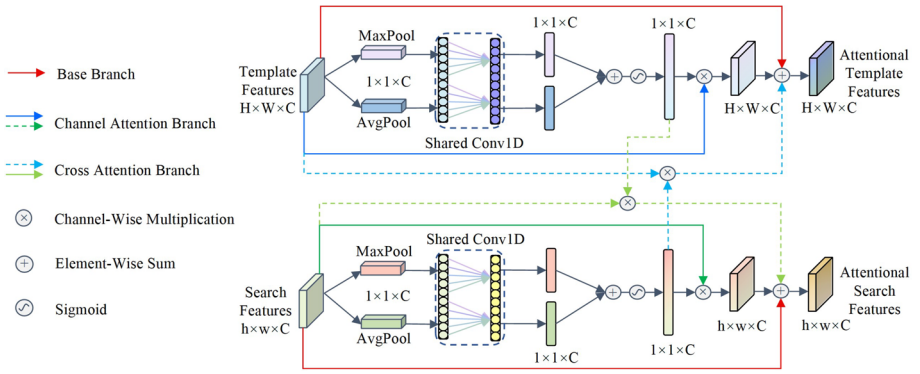

Fig. 3 The Main architecture of Cross-RAAN. The network involves two branches of both template branch and search branch, in which contains three sub-branch base branch, channel attention branch and cross attention branch, where base branch retains the original feature, channel attention branch produces channel feature in response to weights with variable channel and cross attention branch creates cross feature by using Siamese network in cross manner with two branches of channel weights

of the residual structure can effectively enhance the feature representation capability of the network.

To obtain comprehensive feature enhancement, Self-RAAN shares the weights of the network in both branches of the Siamese network, which allows for more efficient exploration of interdependencies between features. By introducing attention in the Siamese backbone, Self-RAAN aggregates and enhances the self-semantic relevance of individual feature graphs to further aggregate different features to provide stable and robust self-attentive features.

3.3 Cross-Residual Attentional Aggregation Network

Siamese network usually conducts prediction using features from the final stage, as the two branches calculating features separately while they share a feature extraction network using the same parameters for calculation, so there are a large number of relevant features waiting to be discovered in both branches. In process of object tracking, it is common for objects to show interferences such as deformation and background clutter. Thus, it is vital for the search branch to learn relevant information about the template image to help generate a more recognizable feature representation, meanwhile, it's possible for template branch to highlight more relevant features in the template image and search target by encoding contextual information from the search branch into the object representation, which facilitates more accurate object localization in various interference situations.

For this purpose, we propose Cross-RAAN, which computes cross-attention on the features of two branches of the Siamese backbone network. By computing cross-channel features, Cross-RAAN is able to capture the correlation between the search branch and the template branch, which can help extract more discriminative features. By interacting features across branches, the network is able to better utilize the correlated features to adapt to various interference situations and improve the robustness of target tracking.

In detailed, the proposed Cross-RAAN embedded at the end of two branches of Siamese backbone network is to perform cross-attention computing separately for the end of three stages features with different depths, as shown in Fig. 3. By sharing attention weights between two Siamese branches in order to learn information from each other, the two branches can

perform more collaboratively in extracting features. Cross-RAAN includes three branches: base, channel attention and cross attention. Where the base branch stores the original information of the features, the channel attention branch is designed to enhance the focus of the image features, and the same idea of channel attention is aided for the cross attention branch, but the weight of attention comes from another branch, so as to complete the collaborative work of the template branch and the search branch in Siamese network to obtain better feature extraction outcomes.

Due to template features and search features with the same number of channels but in spatial size, we try to obtain attention weights based on the same channel attention computing method like the channel attention model of Self-RAAN. Considering the one-dimension applied in channel attention of Cross-RAAN and the different three-level depth features of Siamese backbone network, the same size of the convolutional kernel will alleviate the representation ability of the model, so it's necessary to design a convolutional kernel with different sizes regardless of different depth feature. For the Conv3 and Conv4 layers corresponding to Cross-RAAN, we use a convolution kernel of size 5. Since the number of channels generated by the Conv5 layer is 2048, which greatly exceeds the 512 of the Conv3 layer and 1024 of the Conv4 layer, further application of a convolutional kernel of size 5 will weaken the performance of the model, and we set the size of the convolutional kernel of the Cross-RAAN corresponding to the Conv5 layer to 7. Given the input features of template branch as $Z \in \mathbb{R}^{H\times W\times C}$, the channel attention map as $A_z \in \mathbb{R}^{H\times W\times 1}$, output of cross attention features as $M_z \in \mathbb{R}^{H\times W\times C}$, the input features of search branch as $X \in \mathbb{R}^{h\times w\times C}$, channel attention map as $A_x \in \mathbb{R}^{H\times W\times 1}$ and output of cross attention features as $M_x \in \mathbb{R}^{H\times W\times C}$, respectively. The process of cross-feature enhancement is computed as follows:

$$\begin{aligned} A_z &= \sigma\{f^k(MaxPool(Z)) + f^k(AvgPool(Z))\} \\ A_x &= \sigma\{f^k(MaxPool(X)) + f^k(AvgPool(X))\} \end{aligned} \tag{11}$$

$$\begin{aligned} M_z &= A_z \times Z + A_x \times Z + Z, \\ M_x &= A_x \times X + A_z \times X + X, \end{aligned} \tag{12}$$

where σ presents sigmoid operation and f^k symbols the one-dimension convolutional operation with convolution kernel is equal to k.

In order to maintain synergy between the template branch and the search branch when extracting features, the two branches of the Siamese network share the parameters of the one-dimensional convolution in Cross-RAAN to further enhance the feature representation of both branches. Cross-RAAN aggregates and enhances the semantic relevance of the feature maps of both branches of the Siamese network, highlights the productive information from the complex feature maps, and reduces interference from factors such as occlusion, providing stable and robust cross-attention features for subsequent regression classification networks.

3.4 Multi-Level Feature Fusion Module

Convolutional features of variable depths represent different information, and many methods [16, 33, 34] use fusion of features of different dimensions to improve tracking accuracy. Although the Conv3, Conv4 and Conv5 layers of the backbone network have the same spatial resolution, their atrous convolution has different expansion rates, resulting in significant variations in the feature information captured by the three convolutional layers. CF [35] proposed that among the convolutional layers of different depths, the pre-layer captures fine-grained information, such as edges, color and shape, which are essential for locating the target

location, while the post-layer has a larger number of feature channels, which is more helpful for encoding the abstract semantic information of the target and improves the robustness to interference situations such as changes in object appearance.

The multi-level feature fusion (MFF) module was further developed with the aim of exploiting the most representative information from the correlation features of different depths. This module can effectively fuse features from different depths to enhance the performance of target tracking. By establishing feature connections between the pre and post layers, we can effectively interact and share the fine-grained information from the bottom layer with the semantic information from the top layer. This cross-depth feature fusion can help us make full use of different levels of feature representations in target tracking and improve the robustness under target epistemic changes.

In this paper, the proposed approach is unlike the SiamCAR [3], we try to use directly full feature image instead of compressing the feature map prior to deep correlation. MFF model first leverages depth-wise cross correlations [12] operation for both the end three layer features of Cross-RAAN enhancement $Z_i \in \mathbb{R}^{H\times W\times C}$, i = {3, 4, 5} and $X_i \in \mathbb{R}^{h\times w\times C}$, i = {3, 4, 5}, response map R_3, R_4 and R_5 containing 512, 1024 and 2048 is obtained dependently. Then we use the convolutional kernel with 1×1 size to compress the number of channels of three response maps into 256 to obtain R_i^*, i = {3, 4, 5}. Compressing the channel dimension can significantly reduce the number of parameters and speed up subsequent calculations. Finally, the three response maps were again fused into one using a 1 × 1 convolution kernel, which produced a cross-correlated response map combining shallow and deep features and was used as input to the subsequent classification regression network for bounding box prediction, the formulations are as follows:

$$R_i = Z_i \odot X_i, \mathrm{i} = \{3, 4, 5\} \tag{13}$$

$$R_i^* = f_1^{1\times1}(R_i), \mathrm{i} = \{3, 4, 5\} \tag{14}$$

$$R = f_2^{1\times1}(Cat(R_i^*)), \mathrm{i} = \{3, 4, 5\} \tag{15}$$

where $\odot$ presents channel-wise correlation operation, $f_1^{1\times1}$ and $f_2^{1\times1}$ shows the two-dimensional convolutional operation with convolutional kernel size is equal to 1×1 and Cat shows the splicing operation along the channel dimension of feature graph.

3.5 Ground-Truth and Loss

Throughout our training process, there were fewer sample imbalances due to the small proportion of area occupied by the target and the background in the input search region. Therefore in our research, we have employed the cross-entropy loss for classification purposes, while utilizing the Intersection over Union (IoU) loss for regression tasks. We set the coordinates of the upper-left and lower-right corners of the ground truth bounding box as (x_0, y_0) and (x_1, y_1), the corresponding position of point (i, j) is denoted by (x, y). The regression targets $\widetilde{t}_{i,j}$ at $A_{w\times h\times4}^{reg}(i, j, :)$ can be computed by:

$$\begin{aligned} \widetilde{t}_{i,j}^{0} &= \widetilde{l} = x - x_0, \widetilde{t}_{i,j}^{1} = \widetilde{t} = y - y_0, \\ \widetilde{t}_{i,j}^{2} &= \widetilde{r} = x_1 - x, \widetilde{t}_{i,j}^{3} = \widetilde{b} = y_1 - y. \end{aligned} \tag{16}$$

With $\widetilde{t}_{i,j}$, the IOU can be computed between the ground-truth bounding box and the predicted bounding box. Then we compute the regression loss by using

$$\mathcal{L}_{reg} = \frac{1}{\sum \mathbb{I}(\widetilde{t}_{(i,j)})} \sum_{i,j} \mathbb{I}(\widetilde{t}_{(i,j)}) L_{IOU}(A^{reg}(i,j,:), \widetilde{t}_{(x,y)}), \tag{17}$$

where L_{IOU} is the IOU loss and $\mathbb{I}(\cdot)$ is a custom function defined by:

$$\mathbb{I}(\widetilde{t}_{(i,j)}) = \begin{cases} 1 & \text{if } \widetilde{t}^{k}_{(i,j)} > 0, k = 0, 1, 2, 3, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

The branch parallel to the classification branch is the center-ness branch, which is used to eliminate outliers generated at locations far from the center of the target. The score $C(i, j)$ in $A^{cen}_{w\times h\times 1}(i, j)$ is defined by:

$$C(i,j) = \mathbb{I}(\widetilde{t}_{i,j}) \times \sqrt{\frac{\min(\widetilde{l}, \widetilde{r})}{\max(\widetilde{l}, \widetilde{r})} \times \frac{\min(\widetilde{t}, \widetilde{b})}{\max(\widetilde{t}, \widetilde{b})}}, \tag{19}$$

where $C(i, j)$ is in contrast with the distance between the corresponding location (x, y) and the object center in the search region. If (x, y) is located in the background, the value of $C(i, j)$ is set to 0. The center-ness loss is

$$\begin{aligned} \mathcal{L}_{cen} = \frac{-1}{\sum \mathbb{I}(\widetilde{t}_{(i,j)})} \sum_{\mathbb{I}(\widetilde{t}_{(i,j)})==1} & [C(i,j) \times \log A^{cen}_{w\times h\times 1}(i,j) + \\ & (1 - C(i,j)) \times \log(1 - A^{cen}_{w\times h\times 1}(i,j))]. \end{aligned} \tag{20}$$

The overall loss function is

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{cen} + \lambda \mathcal{L}_{reg} \tag{21}$$

where $\mathcal{L}_{cls}$ represents the cross-entropy loss for classification. During model training, different values of λ can bring different accuracy to regression prediction. We set the constant $\lambda = 3$ based on our experimental experience.

4 Experiments

In this section, our experiments aim to explore, evaluate and validate the effectiveness and performance of our proposed method. To achieve this goal, we designed a series of experiments and collected relevant data for qualitative analysis and comparison. This section will first summarize the implementation details of the experiments, and then step-by-step present the qualitative analysis of the experiments and the test results of the various datasets as well as the ablation experimental part.

4.1 Implementation Details

The proposed SiamRAAN is implemented in Python with Pytorch on a single Tesla V100. The input size of template patch and search region is same as SiamRPN++ [16] with 127×127 and 255×255 pixels respectively. We tend to use backbone networks initialized with parameters pre-trained on ImageNet [36] to extract features.

Our model adopts the Stochastic Gradient Descent (SGD) to train 20 epochs. The warmup learning rate of prior five epochs vary from 0.001 to 0.005, the value of remaining 15 epochs decreases from 0.005 to 0.0005 in exponential decay. In the prior half of epochs, the value

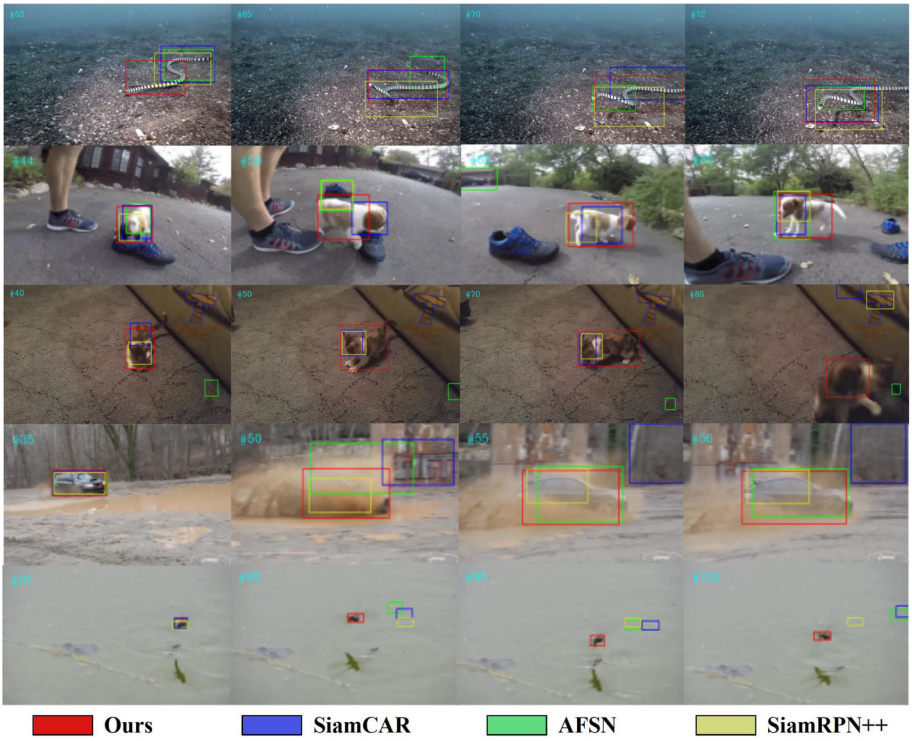

Fig. 4 In comparison with three excellent trackers based on GOT-10K with five challenging sequences. Our network successfully tackles the challenges in accordance with object removing fast and scale changes

of batch size is fixed 80 and the parameters of Siamese backbone network are frozen during training classification and regression network. And in the end of the half, last three-stages of ResNet-50 are release to train and reduce the number of batch size to 32. For overall training, our RAAN and MFF module performs training all time. The training datasets of proposed model is equal to COCO [37], ImageNet DET, ImageNet VID [36], YouTube-BB [38], GOT-10K [39] and LaSOT [40], which are to demonstrate the experimental performance on GOT-10K, LaSOT, OTB-50 [41], OTB-100 [42] and UAV123 [43].

4.2 Qualitative Analysis

We visualize the tracking results based on the proposed SiamRAAN, AFSN, SiamCAR and SiamRPN++ in dataset GOT-10K test to qualitatively evaluate our methods and demonstrate the effectiveness of our algorithms, and choose five video test sequences to evaluate the tracking effectiveness of the proposed network and the object prediction frames for each algorithm are shown in Fig. 4.

In the videos in the first and second rows, where there are interference conditions such as deformation of the target, all the compared algorithms succeed in tracking the target, but our SiamRAAN show the more superiority that it always uses the main body of the target as the tracking object, while the remaining three trackers fail to recognize the deformed target as a whole, which results in the predicted bounding box not covering the target completely.

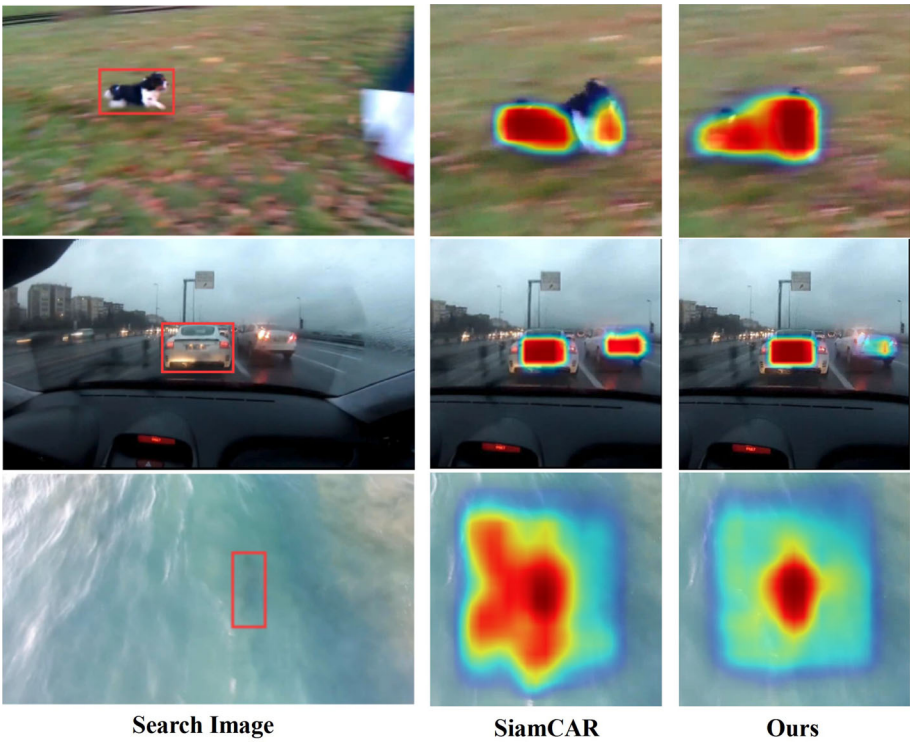

Fig. 5 Visualization of confidence maps used by SiamCAR (middle) and our method (right) SiamRAAN. The red box indicates the tracking object for each image (left). the feature map of SiamCAR does not cope well with the object location. In contrast, the object confidence map obtained by our model has a stronger discriminative power in predicting the target location

In the video in row 3, the target suffers from fast moving and deformation interference. AFSN loses the target in frame 40, SiamCAR and SiamRPN++ only focus on part of the target's position in frames 50 and 70 and cannot identify the overall target profile, while the target is completely lost in frame 85. Our proposed SiamRAAN, by contrast, never lose the target position throughout and is robust to fast-moving targets that produce large amounts of deformation. In the video in row 4, the complex surroundings of the target lead to a drifting situation in the AFSN in frame 50, while SiamCAR fails to identify the target on the whole and the target bounding box deviates from the target. In the video in row 5, there is interference such as fast moving and low resolution in the target, and SiamCAR, AFSN and SiamRPN++ lose the target at frame 92, which leads to the subsequent inability to track the target consistently, making the tracking accuracy lower.

Visualization of confidence maps used by SiamCAR (middle) and our method (right) is to confirm the effectiveness of the proposed algorithm with three video sequences in the LaSOT dataset as shown in Fig. 5.

In the first row, SiamCAR only focuses on the partial features of the target and fails to achieve recognition of the target as a whole when the target with a fast moving and deforming state, while our proposed SiamRAAN identifies the target as a whole. In the second row, in the face of the presence of similar semantic information, SiamCAR identifies the right-hand interference as a target as well and with high confidence. SiamRAAN is not completely

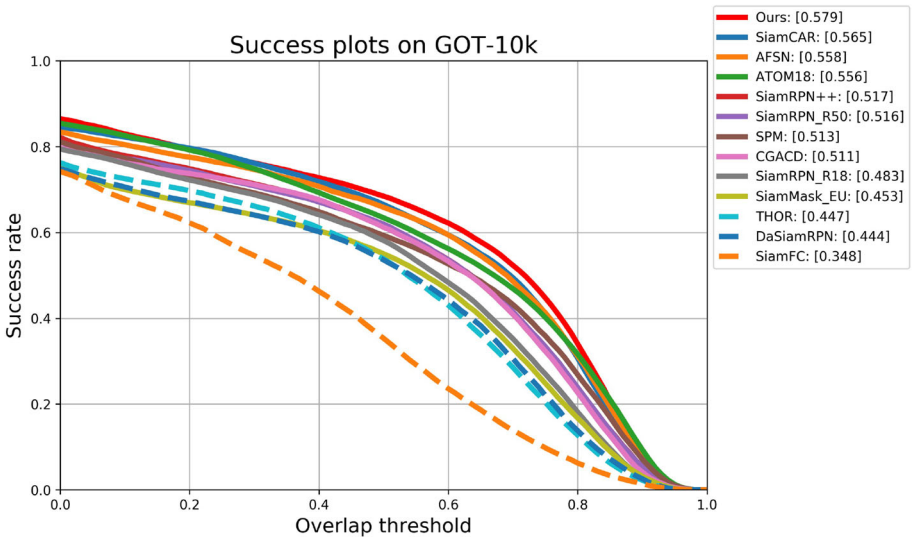

Fig. 6 Experimental results of different methods on GOT-10K testing dataset

excluded from similar interference with low confidence level, which has no impact on tracking normally. In the third row, SiamCAR failed to recognize the outline of the target confronted with blurred target features since it identifies the environment to the left of the target as part of the target as well, while SiamRAAN successfully detect the complete outline of the target. In summary, with the analysis of multiple video sequences, the proposed object tracking network in this paper provides better accuracy and robustness in the face of various challenging tasks.

4.3 Results on GOT-10K

GOT-10K contains more than 10000 video sequences of objects moving in the real world. The comparison results are shown in Fig. 6 and Table. 1 in accordance with the proposed tracker and other trackers. Evaluation indicators including success plots, average overlap (AO) and success plots (SR) are to verify the performance, where AO indicates the average overlap between all estimated bounding boxes and the true bounding box. $SR_{0.5}$ presents the percentage of successfully tracked frames with an overlap greater than 0.5, while $SR_{0.75}$ shows the percentage of successfully tracked frames with an overlap greater than 0.75. We evaluate the performance of SiamRAAN on GOT-10K dataset in comparison with SiamCAR [3], SiamRPN++ [16], SPM [34] and other nine cutting-edge baseline trackers. Compared with SiamCAR, the proposed SiamRAAN enables to improve AO, $SR_{0.5}$ and $SR_{0.75}$ by an average of 1.4%, 2.0% and 3.5%, respectively.

4.4 Results on LaSOT

Dataset LaSOT contains over 3.25 million manually annotated frames and 1,400 videos with covering a total of 70 categories in 20 tracking sequences. Evaluation indicators based on the dataset including precision plots and success plots in one-pass evaluation (OPE) are to

Table 1 Evaluation on GOT-10K testing dataset

Tracker	AO	$SR_{0.5}$	$SR_{0.75}$	FPS	Hardware	Language
Ours	**0.579**	**0.685**	**0.447**	51.62	Tesla V100	Python
SiamCAR [3]	0.565	0.665	0.412	55.89	Tesla V100	Python
AFSN	0.558	0.659	0.413	39.06	GTX 1080ti	Python
ATOM18	0.556	0.634	0.402	20.71	GTX 1050	Python
SiamRPN++ [16]	0.517	0.616	0.325	49.83	RTX 2080ti	Python
SiamRPN_R50 [14]	0.516	0.620	0.334	26.68	GTX 1080Ti	Python
SPM [34]	0.513	0.593	0.359	72.30	Titan Xp	Python
CGACD [27]	0.511	0.612	0.323	37.73	Tesla P100	Python
SiamRPN_R18	0.483	0.581	0.270	97.55	Titan X	Python
SiamMask_EU	0.453	0.550	0.248	15.37	Tesla P100	Python
THOR	0.447	0.538	0.204	1.00	RTX 2070	Python
DaSiamRPN [15]	0.444	0.536	0.220	134.40	Titan RTX	Python
SiamFC [12]	0.348	0.353	0.097	44.52	Titan X	Python

The best results are in bold
The trackers are ranked according to **AO** measure

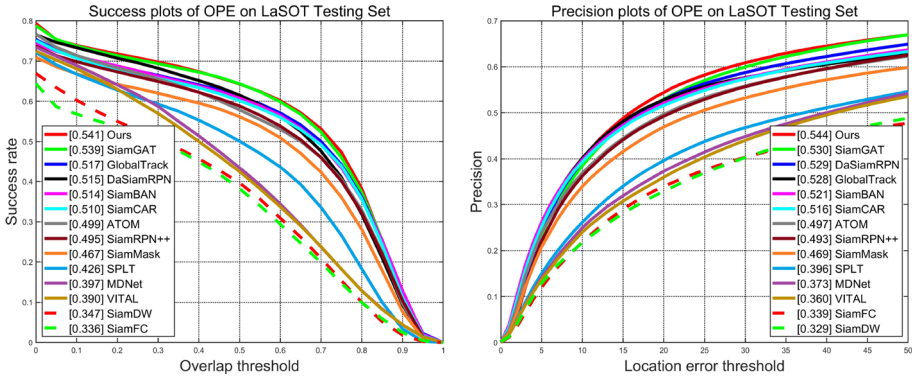

Fig. 7 Results OPE on LaSOT

verify the performance. We tend to validate the performance of SiamRAAN compared with 13 cutting-edge baseline trackers such as DaSiamRPN [15], SiamGAT [25], SiamCAR [3] and SiamRPN++ [16] et al. The proposed model enables to improve the OPE and precision plots by an average of 3.1% and 2.8% contrasted with SiamCAR as shown in Fig. 7.

4.5 Results on OTB-50

OTB-50 Contains 50 challenging videos with large variations and considers the average success rate per frame at different thresholds. A tracker is considered successful for a given frame if the intersection and concurrency ratio (IoU) between the predicted and true values of the tracker is higher than a certain threshold. These trackers are then compared based on the area under the success rate curve at different thresholds. We compare our SiamRAAN with a set of trackers including SiamCAR [3], TADT [44], DaSiamRPN [15], SiamRPN [14]

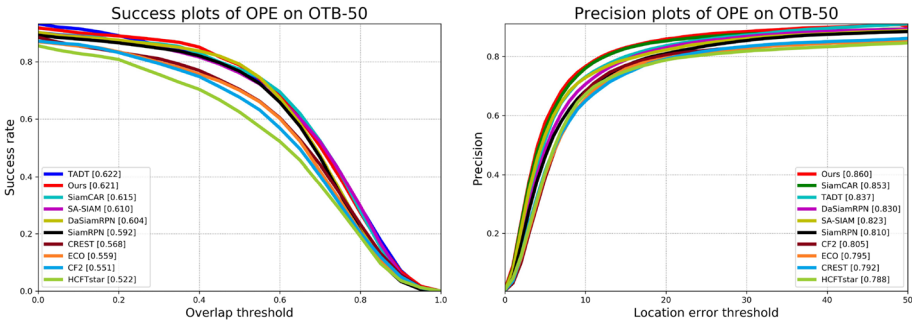

Fig. 8 Results OPE on OTB-50

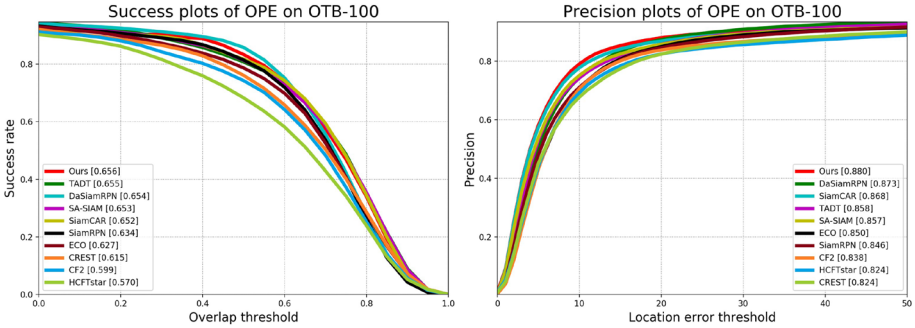

Fig. 9 Results OPE on OTB-100

and nine other baseline methods. Figure 8 shows the results of our tracker compared to the 9 state-of-the-art trackers. Compared with SiamCAR, SiamRAAN improved the success rate and accuracy by 0.6% and 0.7%, respectively.

4.6 Results on OTB-100

The OTB-100 dataset is a widely used in tracking benchmark containing 100 challenging videos. We attempt to confirm the performance of SiamRAAN contrasted with 9 baseline trackers such as SiamCAR [3], TADT [44], DaSiamRPN [15] et al. as presented in Fig. 9 shows the experimental results. In comparison with SiamCAR, SiamRAAN could improve the success rate and precision rate by an average of 0.4% and 1.2% respectively.

4.7 Results on UAV123

The UAV123 dataset contains 123 video sequences and over 110K frames. All sequences are fully labeled with upright borders. Objects in the dataset can be seen with fast motion, large scale and lighting variations, and occlusion, which makes tracking with this dataset challenging. We compare our SiamRAAN with state-of-the-art trackers including SiamCAR [3], SiamGAT [25], SiamRPN [14] and SiamRPN++ [16]. Here, we use the success rate curve and accuracy rate curve of OPE to evaluate the overall performance. Figure 10 shows the results of our tracker compared to the 9 state-of-the-art trackers. Compared to SiamCAR,

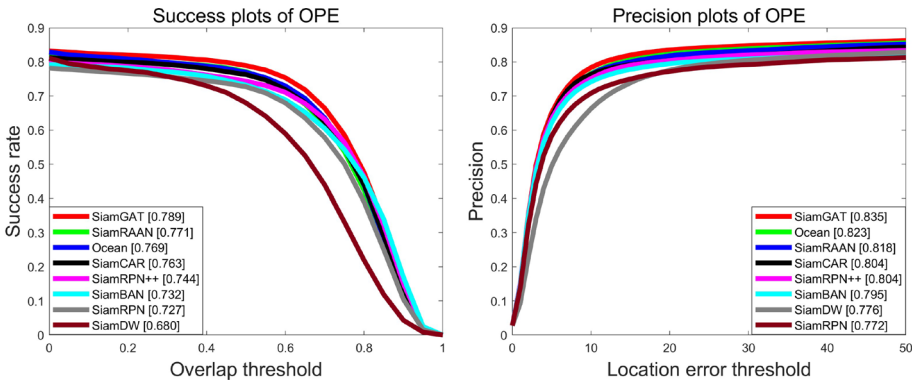

Fig. 10 Results OPE on UAV123

the success rate and accuracy rate of our SiamRAAN are improved by 0.8% and 1.4%, respectively.

4.8 Attribute-Based Comparison

The test sequences of LaSOT dataset contains 14 properties to express multiple challenging aspects including illumination variation, partial occlusion, deformation, motion blur, camera motion, rotation, background clutter, viewpoint change, scale variation, full occlusion, fast motion, out-of-view, low resolution and aspect ration change. Compared with 13 cutting-edge methods such as DaSiamRPN [15], SiamGAT [25], SiamCAR [3] and SiamRPN++ [16] et al. to demonstrate the performance of the proposed model and Fig. 11, Fig. 12 show the experimental outcomes of our method and other trackers conducted on multiple different attributes. The overall performance of SiamRAAN ranks among the first or second in precision rate and success rate. The proposed model with superior performance in the face of changes in target aspect ratio and deformation attributes to both the proposed Self-RAAN will capture more detailed information connected with the object greatly and our Cross-RAAN can enhance effectively object template feature. Especially in the case of background clutter, the proposed mechanism enables to improve the precision rate and success rate to 6.1% and 4.8% with the help of RAAN and MFF modules.

4.9 Ablation Study

In this subsection, we try to perform an ablation analysis of the proposed Residual attentional aggregation network (RAAN) and (MFF) modules on the GOT-10K dataset. Noted that in the table. 2, our method can improve the score of AO, $SR_{0.5}$ and $SR_{0.75}$ by 0.3%, 0.6% and 2.6% on GOT-10K dataset when we firstly added Self-RAAN module, the approach will increase the score of evaluation metrics by 0.5%, 0.7% and 0.5% as added the Cross-RAAN module subsequently, and finally the framework enables to increase the score of these indicators by 0.6%, 0.7% and 0.4% after embedded with MFF module. Meanwhile, it's noticed that our baseline algorithm SiamCAR ran at 55.89 FPS while our method SiamRAAN ran at 51.62 FPS when tested under the same experimental conditions, which demonstrates that the

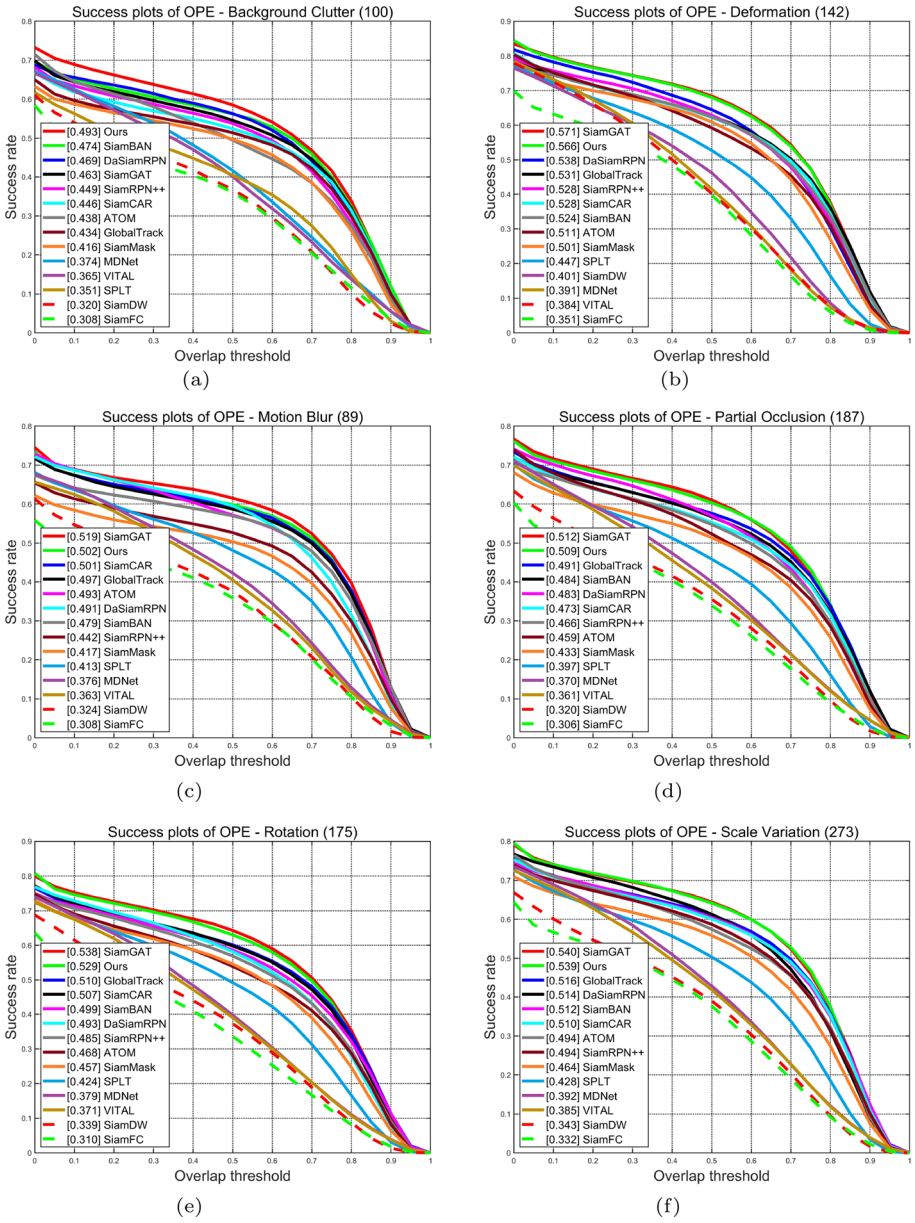

Fig. 11 The success plots for six challenging attributes: background clutter(a), deformation(b), motion blur(c), partial occlusion(d), rotation(e) and scale variation(f) on LaSOT testing dataset

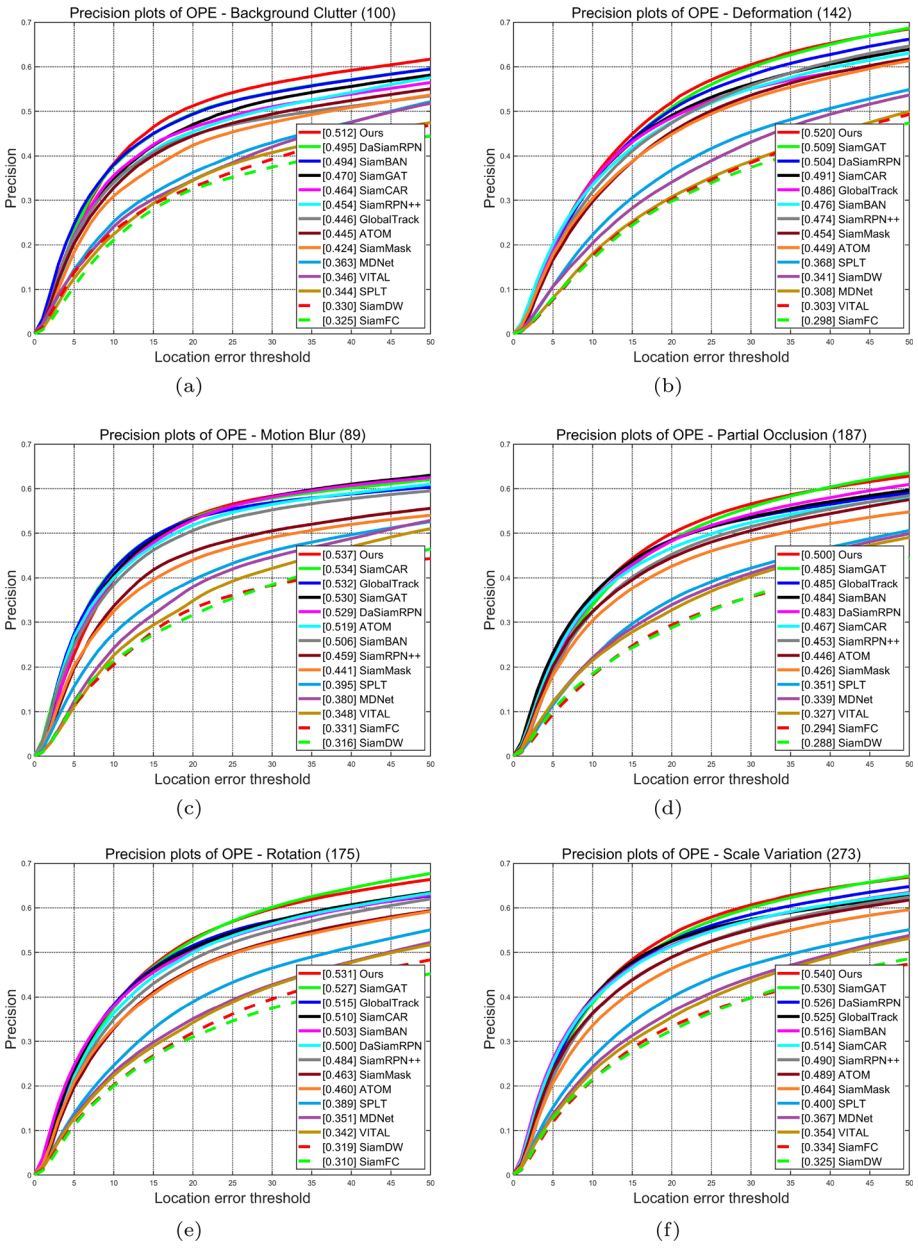

Fig. 12 The precision plots for six challenging attributes: background clutter(a), deformation(b), motion blur(c), partial occlusion(d), rotation(e) and scale variation(f) on LaSOT testing dataset

Table 2 A Comparison among different modules based on GOT-10K testing dataset under the same experimental conditions

Self-RAAN	Cross-RAAN	MFF	GOT-10K			
			AO	$SR_{0.5}$	$SR_{0.75}$	FPS
			0.565	0.665	0.412	55.89
✓			0.568	0.671	0.438	53.63
✓	✓		0.573	0.678	0.443	52.95
✓	✓	✓	0.579	0.685	0.447	51.62

RAAN and MFF modules perform without a significant burden on the tracking speed while achieving higher accuracy and robustness.

5 Conclusion

This paper proposes a novel Siamese Residual Attentional Aggregation Network (SiamRAAN) framework, based on the thought of feature enhancement, in combination with the features of Siamese network with two branches simultaneously. The framework enables to effectively leverage the its features information of channel and spatial and to realize robustness tracking in a complementary way with two branches information in Siamese network. In comparison with baseline SiamCAR, the proposed tracker has significant improvement conducted on the datasets GOT-10K, LaSOT, OTB-50, OTB-100 and UAV123 and demonstrates its effectiveness.

Since our model uses multi-layer attention for feature enhancement, the training and inference of SiamRAAN are slower than normal Siamese methods, while requiring more GPU resources. Also, SiamRAAN may have difficulty in accurately distinguishing between tracking targets and environmental objects when they are extremely similar in appearance. In addition, SiamRAAN always uses the first frame as the input to the template branch during tracking, and lacks the integration of historical information, making it difficult to cope with severe distortion of appearance in long-term sequences. In the future, we plan to further improve our tracker in terms of model compression, inter-branch correlation feature enhancement, history information fusion, etc.

Acknowledgements This work was supported by the Key Science and Technology Project of Henan Province (Grant No. 201300210400), and Henan Province Science and Technology Research Project (Grant No. 232102210031).

Author Contributions ZX: Data curation, Software, Writing—original draft, Methodology. JY: Conceptualization, Supervision. XH: Investigation, Software, Validation. YS: Methodology, Writing—review and editing. HL: Supervision, Writing review and editing.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Held D, Thrun S, Savarese S (2016) Learning to track at 100 fps with deep regression networks. In: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, 11–14 Oct 2016, Proceedings, Part I 14, pp 749–765. Springer
2. Henriques JF, Caseiro R, Martins P, Batista J (2014) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
3. Guo D, Wang J, Cui Y, Wang Z, Chen S (2020) Siamcar: siamese fully convolutional classification and regression for visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6269–6277
4. Wang F, Cao P, Wang X, He B, Sun F (2023) Siamadt: siamese attention and deformable features fusion network for visual object tracking. Neural Process Lett, pp 1–18
5. Wu Y, Cai C, Yeo CK (2023) Siamese centerness prediction network for real-time visual object tracking. Neural Process Lett 55(2):1029–1044
6. Yu J, Rui Y, Tao D (2014) Click prediction for web image reranking using multimodal sparse coding. IEEE Trans Image Process 23(5):2019–2032
7. Yu J, Tan M, Zhang H, Rui Y, Tao D (2019) Hierarchical deep click feature prediction for fine-grained image recognition. IEEE Trans Pattern Anal Mach Intell 44(2):563–578
8. Zhang J, Cao Y, Wu Q (2021) Vector of locally and adaptively aggregated descriptors for image feature representation. Pattern Recogn 116:107952
9. Zhang J, Yang J, Yu J, Fan J (2022) Semisupervised image classification by mutual learning of multiple self-supervised models. Int J Intell Syst 37(5):3117–3141
10. Marvasti-Zadeh SM, Cheng L, Ghanei-Yakhdan H, Kasaei S (2021) Deep learning for visual tracking: A comprehensive survey. IEEE Trans Intell Transp Syst
11. Wu X, Sahoo D, Hoi SC (2020) Recent advances in deep learning for object detection. Neurocomputing 396:39–64
12. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH (2016) Fully-convolutional siamese networks for object tracking. In: Computer vision–ECCV 2016 workshops: Amsterdam, The Netherlands, 8–10 and 15–16, Oct 2016, Proceedings, Part II 14, pp 850–865. Springer
13. Liu Q, Li X, He Z, Fan N, Yuan D, Wang H (2020) Learning deep multi-level similarity for thermal infrared object tracking. IEEE Trans Multimedia 23:2114–2126
14. Li B, Yan J, Wu W, Zhu Z, Hu X (2018) High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8971–8980
15. Zhu Z, Wang Q, Li B, Wu W, Yan J, Hu W (2018) Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European conference on computer vision (ECCV), pp 101–117
16. Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J (2019) Siamrpn++: evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4282–4291
17. Fan H, Ling H (2019) Siamese cascaded region proposal networks for real-time visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7952–7961
18. Guo Q, Feng W, Zhou C, Huang R, Wan L, Wang S (2017) Learning dynamic siamese network for visual object tracking. In: Proceedings of the IEEE international conference on computer vision, pp 1763–1771
19. Yang T, Chan AB (2018) Learning dynamic memory networks for object tracking. In: Proceedings of the European conference on computer vision (ECCV), pp 152–167
20. Wang Q, Teng Z, Xing J, Gao J, Hu W, Maybank S (2018) Learning attentions: residual attentional siamese network for high performance online visual tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4854–4863
21. Yu J, Zuo M, Dong L, Zhang H, He X (2022) The multi-level classification and regression network for visual tracking via residual channel attention. Digital Signal Process 120:103269
22. Tao R, Gavves E, Smeulders AW (2016) Siamese instance search for tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1420–1429

23. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28
24. Zhang Z, Peng H (2019) Deeper and wider siamese networks for real-time visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4591–4600
25. Guo D, Shao Y, Cui Y, Wang Z, Zhang L, Shen C (2021) Graph attention tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9543–9552
26. Zhao Z, Zuo M, Yu J, He X, Song Y, Zhai R (2022) Siamese network based on global and local feature matching for object tracking. J Electronic Imag 31(6):063022
27. Du F, Liu P, Zhao W, Tang X (2020) Correlation-guided attention for corner detection based visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6836–6845
28. Yu Y, Xiong Y, Huang W, Scott MR (2020) Deformable siamese attention networks for visual object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6728–6737
29. Yang S, Chen H, Xu F, Li Y, Yuan J (2022) High-performance uavs visual tracking based on siamese network. Visual Comput, pp 1–17
30. Guo C, Yang D, Li C, Song P (2022) Dual siamese network for rgbt tracking via fusing predicted position maps. Visual Comput 38(7):2555–2567
31. Pang H, Han L, Liu C, Ma R (2023) Siamese object tracking based on multi-frequency enhancement feature. Visual Comput, pp 1–11
32. Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp 3–19
33. Ma C, Huang J-B, Yang X, Yang M-H (2018) Robust visual tracking via hierarchical convolutional features. IEEE Trans Pattern Anal Mach Intell 41(11):2709–2723
34. Wang G, Luo C, Xiong Z, Zeng W (2019) Spm-tracker: Series-parallel matching for real-time visual object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3643–3652
35. Ma C, Huang J-B, Yang X, Yang M-H (2015) Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE international conference on computer vision, pp 3074–3082
36. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115:211–252
37. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European conference, Zurich, Switzerland, 6–12 Sept 2014, Proceedings, Part V 13, pp 740–755. Springer
38. Real E, Shlens J, Mazzocchi S, Pan X, Vanhoucke V (2017) Youtube-boundingboxes: a large high-precision human-annotated data set for object detection in video. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5296–5305
39. Huang L, Zhao X, Huang K (2019) Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Trans Pattern Anal Mach Intell 43(5):1562–1577
40. Fan H, Lin L, Yang F, Chu P, Deng G, Yu S, Bai H, Xu Y, Liao C, Ling H (2019) Lasot: a high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5374–5383
41. Wu Y, Lim J, Yang M-H (2013) Online object tracking: A benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2411–2418
42. Wu Y, Lim J, Yang M (2015) Object tracking benchmark. IEEE Trans Pattern Anal Mach Intell
43. Mueller M, Smith N, Ghanem B (2016) A benchmark and simulator for uav tracking. In: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, 11–14 Oct 2016, Proceedings, Part I 14, pp 445–461. Springer
44. Li X, Ma C, Wu B, He Z, Yang M-H (2019) Target-aware deep tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1369–1378