ORIGINAL ARTICLE
Journal Section

Robust Siamese Tracking with Asymmetrical Feature Processing Network

Zhongjie Mao[1] | **Xi Chen**[1*] | **Jia Yan**[2] | **Shenghua Fan**[1]

[1]School of Computer Science, Wuhan University, China

[2]Department of Electrical Engineering, School of Electronic Information, Wuhan University, China

Correspondence
Xi Chen, School of Computer Science, Wuhan University, China
Email: robertcx@whu.edu.cn

Funding information
Funder One, National Natural Science Foundation of China, Grant/Award Number: 61501334

The Siamese tracker consists of two components: a classification and a regression networks. Despite their different roles, most Siamese trackers have similar feature fusion modules in the two networks, leading to the neglect of their unique characteristics. In this work, we experimentally discover that the two networks place different levels of emphasis on different types of information. Specifically, regression tends to rely on semantic information, while classification places more emphasis on global information. Therefore, we propose a new tracking structure named SGTrack, which includes a semantic augmentation fusion (SAF) for regression and a global relevance fusion (GRF) for classification. It allows us to unlock the full potential of both networks. The experimental results of our method on five benchmarks provide evidence of a notable improvement in tracking performance, while preserving real-time speed.

KEYWORDS
Object tracking, Siamese network, Semantic Augmentation, Global Relevance

1 | INTRODUCTION

Visual object tracking is widely recognized as an essential task within the computer vision domain [1]. For the single object tracking, given a ground-truth bounding box in the first frame as the prior knowledge, the target position should

be predicted in subsequent frames. Significant progress has been achieved in the past years in the object tracking field by utilizing deep neural networks as a powerful appearance model. Recently, methods based on Siamese networks [2, 3, 4, 5, 6, 7, 8] have received much attention due to their desirable balance between performance and speed. In the Siamese approaches, visual tracking is approached as a matching problem that involves training two Siamese branches to extract features of the target template and search region, respectively. After that, similarity measurement between two image regions is determined using cross-correlation. The tracking process is then carried out by identifying the image region that bears the highest similarity to the target template in the prediction head. Because Siamese trackers abandon the online learning paradigm and simplify the tracking process, it can be trained end-to-end via annotated images and unleash the potential of big data.

The pioneering work of Siamese trackers is SINT [2], which is a matching-based method with optical flow as motion information. SiamFC [3] takes advantage of a fully convolutional Siamese network to learn a similarity function between the target and search region. Recently, SiamFC has been extended in various ways to improve tracking performance.. Li et al. [9] propose a tracking framework which integrates discriminative correlation filter and SiamFC, exploiting their complementarity. SiamRPN [4] proposes the integration of a regression branch into the tracking process with the goal of enhancing the accuracy of tracking outcomes by extracting region proposals. Pi et al. [10] propose an adaptive feature fusion method to obtain the discriminative high-resolution feature in the Siamese network. SiamRPN++ [5] successfully trained Siamese trackers with deeper backbone. Ocean [6], SiamBAN [7], and SiamFC++ [8] introduce the anchor-free paradigm to object tracking, which directly regresses the location. Jiang et al. [11] build up a Siamese network to mine the dependencies between the template and the search branches. Fan et al. [12] propose an effective feature alignment module and aggregation module for Siamese networks.

Despite the outstanding success of Siamese trackers, their distinct characteristics between the classification and regression tasks are often overlooked. It is reflected in two aspects: (1) many trackers use similar structures for preprocessing features in the classification and regression tasks, instead of properly handling the features according to the characteristics of the different tasks; (2) most trackers use cross-correlation to fuse features in the two tasks, instead of using different fusion strategies. In this work, we experimentally discover that the two networks place different levels of emphasis on different types of information. Specifically, regression tends to rely on semantic information, while classification places more emphasis on global information. Therefore, we propose a novel tracker named SGTrack which exploits asymmetric fusion structures to directly addresses the aforementioned limitations, as shown in Figure 1. Firstly, we argue that the regression task tends to focus on the target itself since target semantic information contains some prior knowledge about the target's appearance, such as color, shape or ratio. The target semantic information is therefore more helpful to estimate scale. We utilize a semantic augmentation fusion (SAF) to establish the linkage between the target's semantic and its appearance, guiding the regression network for more accurate estimation. Secondly, we argue that the classification task needs to exploit more context information since the hard negative examples introduced by the context help prevent overfitting to the easy background. We thus introduce a global relevance fusion (GRF) to capture global information and reinforce the features for the target and the search area, and establish long-range associations between these features. This is able to provide high robustness against distractor objects for classification.

The intuition of the SAF module is that channel features can reveal the target semantic[4, 13]. Some channels have high responses to the target in a specific category, while other channels have negligible responses. SAF module first employs channel-wise attention to generate the weighting coefficients of channels for the target features, which encode the importance of the various channels and represents the target semantic. Then, these coefficients are regarded as the selector to reweight different channels for the target feature map and the search area feature map, enhancing target-specific channels for the two feature maps. In order to capitalize on additional semantic information, a depth-

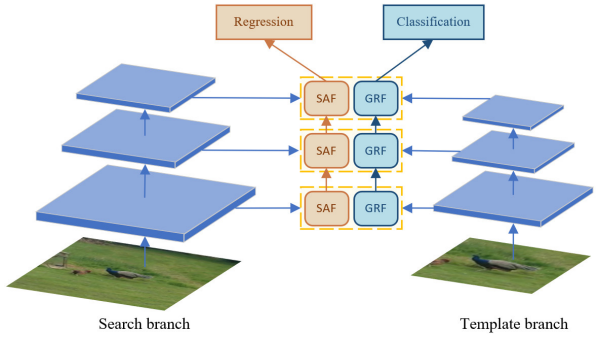

FIGURE 1 In the previous Siamese network, the feature fusion modules of classification and regression branches adopt similar structures. The proposed SGTrack network adopts asymmetric fusion structures according to the different characteristics between the classification and regression tasks.

wise cross-correlation is employed to merge these maps. The final fused feature contains the target appearance and target-specific semantic information. The regression network is therefore able to learn the relationship between these information. Besides, the GRF module exploits a transformer block to process and fuse feature maps since attention in transformer specializes in capturing global information. The module consists of an encoder block and a decoder block. The encoder with self-attention reinforces the target features with global contextual information. The decoder contains a self-attention and a cross-attention modules. The self-attention in decoder is responsible for strengthening the global correlation of the search feature map. The cross-attention is used to mix the target features and the search features, aiming to adaptively focuses on useful information and establishes associations between distant features.

A series of experiments are conducted on OTB100, LaSOT, ITB, UAV20L and UAV123 datasets to prove the generality and effectiveness of the proposed SGTrack. Notably, taking SiamRPN++ as the baseline tracker, our method achieves a substantial improvement with 5.7 points gains on ITB and 2.4 points gains on LaSOT.

The present work can be characterized by the following contributions:

1. We experimentally discover that the two networks place different levels of emphasis on different types of information. Specifically, regression tends to rely on semantic information, while classification places more emphasis on global information. Hence we propose a new tracking structure named SGTrack to unleash respective potentials of the two networks.
2. We exploit a SAF module to inject target semantic into the regression branch, which relates the semantic and appearance of the target and guides the regression network for more accurate estimation. Besides, we use a GRF module to capture global information model long-range dependencies for effectively mixing the target and search area information, providing high robustness against distractor objects for classification.
3. Extensive experiments on five benchmark datasets demonstrate that the proposed tracker significantly improves tracking performance while running at per-frame real-time speeds.

2 | RELATE WORK

This section provides an overview of the related work to our proposed approach, as well as describe the differences with them.

2.1 | Siamese Tracking

Recently, Siamese visual tracking [2, 4, 14, 15, 5, 16] have received much attention due to its tradeoff between accuracy and speed. It treats the tracking task as a matching problem and trains a Siamese network including a target template and search region branches. Following feature extraction in the Siamese network, a cross-correlation computation is employed to determine the similarity between two image regions. Subsequently, the tracking operation entails identifying the image region that exhibits the highest similarity with the target template in the prediction head. Due to the abandonment of the online learning paradigm and the simplification of the tracking process, Siamese trackers can be trained end-to-end via annotated images, thereby unleashing the potential of big data.

SiamFC [2] takes Alex as the backbone to extract features and first introduces cross-correlation as the similarity function for matching. The obtained response is a similarity map for target location prediction, which improves tracking performance. Due to the power of deep learning, this method significantly improves tracking performance. However, the scale estimation with fixed ratios limits the further promotion of SiamFC. In order to solve this problem, Li et al. [4] leverages the region proposal network and adds a region proposal network to get more accurate bounding boxes, which is end-to-end offline trained with large-scale image pairs. Li et al. [9] propose a tracking framework RCT based on SiamFC, which takes advantage of the merits of correlation filters and SiamFC. Similar to SiamFC, RCT uses convolution to extract high-level semantic features and uses correlation filters to refine the location. DaSiamRPN [14] designs a distractor-aware module to achieve more discriminative features and explicitly suppress distractors. Based on SiamRPN, DaSiamRPN [14] designs a distractor-aware module to perform incremental learning and achieves much more discriminative features against semantic distractors. C-RPN [15] proposes to cascade a sequence of RPNs to efficiently exploit high-level and low-level semantic, tackling the problem of data imbalance. Through an innovative sampling strategy, SiamRPN++ [5] effectively addresses the spatial invariance constraint and achieves the successful training of deeper networks. Moreover, SiamRPN++ incorporates a depth-wise cross-correlation layer to combine features from both the template and search branches, thereby producing multiple similarity maps that correspond to different semantic connotations. In general, these RPN-based trackers have achieved great success on performance. It is summarized as follows: 1)optimizing the backbone for better feature representation; 2)replacing match operation for reducing the computational cost; 3)designing head for a more effective bounding box. In this work, we use the SiamRPN++ tracker as the baseline, due to its performance and speed. Differently, we adopt asymmetric fusion structures according to the different characteristics between the classification and regression tasks.

2.2 | Attention Mechanism

Visual attention, as a fundamental component of neural networks, has the potential to accentuate the significant visual areas of interest. SENet [17] design a SE block to improve the representational power of a network by modeling dynamic channel-wise attention. Vaswani et al. [18] propose the Transformer for the machine translation task. The basic block in a transformer is the attention module, which takes a sequence as the input and scans through each element in the sequence and learns their dependencies, aggregating the global information from the input sequence. In order to model long-range dependencies between any two image pixels, the non-local network [19] is employed, with no consideration given to their spatial distance. There are several works that have successfully applied attention to object tracking and achieved performance improvement. ACFN [20] develops an attentional mechanism that chooses a subset of the associated correlation filters for tracking. SA-Siam [21] utilizes a twofold Siamese network to train a semantic branch with integrating channel attention and an appearance branch. RASNet [22] introduces spatial attention and channel attention mechanisms to enhance the discriminative capacity of the deep model. HASiam [23] develops

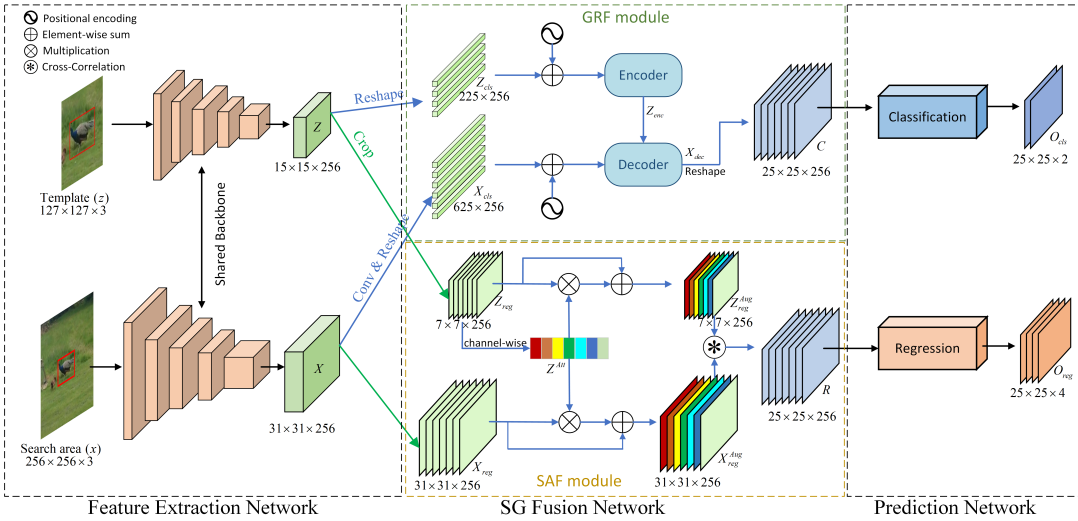

FIGURE 2 Framework of our SGTrack whichcontains three components: a feature extraction network, a SG fusion network which contains the proposed two fusion modules (GRF module and SAF module) and a prediction network which contains two branches (classification and regression). The GRF module is used to pre-process features for the classification branch and the GRF module is used to pre-process features for the regression branch.

a hierarchical attention Siamese network for visual tracking. Gao et al. [24] exploits a novel hierarchical attentional module with long short-term memory and multi-layer perceptrons. The Nocal-Siam [25] approach integrates a target-aware non-local block and a location-aware non-local block to capitalize on long-range dependencies and associate multiple response maps. In this work, we exploit channel attention to extract the target semantic information. Different from the previous work, we inject the semantic information into the target features and search region features simultaneously, aiming to enhance target-specific semantic information in the regression branch. In addition, in the classification branch, we exploit a transformer block to process feature maps since attention in transformer specializes in capturing global information. Differently, we introduce the cross-attention to establish associations between distant features.

3 | PROPOSED METHOD

In this section, we present a detailed description for the proposed SGTrack framework as shown in Figure 2. In specific, we first review the popular Siamese baseline Trackers. Moreover, we describe the proposed SAF module for regression and GRF module for classification. Afterward, we illustrate the proposed tracking framework.

3.1 | Revisit Siamese Framework

SiamFC. We first review this fully-convolutional Siamese tracker. SiamFC consists of two branches which share the same CNN parameters (presented by $\varphi(\cdot)$). It takes a target image z and a candidate search image x as input. The CNN is utilized to extract features of z and x. The shared weight ensures that z and x undergo the same transformation,

which is vital for similarity computing. This yields feature maps, which are cross-correlated as

$$f(z, x) = g(\varphi(z), \varphi(x)), \tag{1}$$

where $g(\cdot)$ denotes a cross-correlation operation. The convolution feature embedding is represented by $\varphi(\cdot)$, while the similarity score map of paired inputs is denoted as $f(\cdot)$. By identifying the maximum score in the response map, the target location can be estimated.

Anchor-based tracking. While the force scale search in SiamFC to obtain target regression is insufficient in accuracy. SiamRPN applied region proposal networks to complement themselves. The RPN network first needs to preset anchor boxes of different sizes on the final map. Then, the RPN framework receives $\varphi(z)$ and $\varphi(x)$ as input and generates dense response maps representing the predicted offsets of all anchors. The scale estimation of each position in the final map is calculated as:

$$\begin{aligned} x &= x_{an} + d_x * w_{an}, \\ y &= y_{an} + d_y * h_{an}, \\ w &= w_{an} + e^{d_w}, \\ h &= h_{an} + e^{d_h}, \end{aligned} \tag{2}$$

where $(x_{an}, y_{an}, w_{an}, h_{an})$ is the preset bounding box of corresponding anchor, (d_x, d_y, d_w, d_h) is the predict offset of corresponding anchor, and (x, y, w, h) is the predicted bounding box of corresponding anchor. Post-processing or filtering is used to select the best bounding box with the best position. In addition, SiamRPN++ uses a depthwise cross-correlation for fusion. Moreover, it exploits multilayer features to predict the target more accurately. We build our tracker based on the SiamRPN++ framework due to its satisfactory balance between performance and efficiency.

3.2 | Semantic Augmentation Fusion

As discussed previously, every channel map usually associated with a particular object class. Some channels have high responses to the target in a specific category, while other channels have negligible responses. SAF module first employs channel-wise attention to generate the weighting coefficients of channels for the target features, which encode the importance of the various channels and represents the target semantic. Besides, the Siamese networks entail the search branch sharing the same backbone network with the template branch, which facilitates the features of both branches undergoing a uniform transformation, leading to highly overlapping identical semantics in certain channels. Thus, these obtained coefficients are regarded as the selector to reweight different channels for the target feature map and the search area feature map, enhancing target-specific channels for the two feature maps. In this way, the target semantic is successfully injected into the regression branch. Additionally, through the back-propagation of regression, the network can acquire the ability to discern the correlation between the target's semantic information and its visual appearance. This constraint helps the tracker to obtain better accuracy.

The features of the target and the search area are respectively represented as $Z \in \mathbb{R}^{C \times H_Z \times W_Z}$ and $X \in \mathbb{R}^{C \times H_X \times W_X}$. Z is first passed through a compression operation, which produces a channel descriptor $Z^{avg} \in \mathbb{R}^{C \times 1 \times 1}$ by aggregating feature maps across their spatial dimensions. This is achieved by using global average pooling to generate

channel-wise statistics, which can be represented by

$$Z^{avg} = \frac{1}{H_Z \times W_Z} \sum_{i=1}^{H_Z} \sum_{j=1}^{W_Z} Z\langle i,j \rangle \tag{3}$$

Where Z^{avg} denotes the channel information of the target image. To make use of this information, a fc (fully-connected) layer and a sigmoid function are followed by the compression operation, which aims to fully capture channel-wise dependencies. The fc layer is capable of learning a nonlinear interaction between channels and the sigmoid is used to ensure that multiple channels are allowed to be emphasized. It can be written as:

$$Z^{Att} = sigmoid(fc(Z^{avg})) \tag{4}$$

where $sigmoid$ is the Sigmoid function, fc is the fully-connected layer. $Z^{Att} \in \mathbb{R}^{C\times1\times1}$ is the channel attention vector, which denotes the importance of each channel of the target. After obtaining the information of C channels, the semantic augmentation feature maps are calculated as:

$$\begin{cases} Z^{Aug}_{i,:,:} = Z_{i,:,:} + Z_{i,:,:} \cdot Z^{Att}_i \\ X^{Aug}_{i,:,:} = X_{i,:,:} + X_{i,:,:} \cdot Z^{Att}_i \end{cases} s.t.\ i \in \{0, 1, ..., C-1\} \tag{5}$$

In which $Z^{Aug}_{i,:,:} \in \mathbb{R}^{1\times H_Z\times W_Z}$ and $X^{Aug}_{i,:,:} \in \mathbb{R}^{1\times H_X\times W_X}$ denote the semantic augmented feature map of i-th channel in the target branch and search branch, respectively. $Z^{Aug} \in \mathbb{R}^{C\times H_Z\times W_Z}$ and $X^{Aug} \in \mathbb{R}^{C\times H_X\times W_X}$ denote the augmented feature maps of all channel, respectively. Due to its tendency to suppress context features and prioritize target-specific semantic information, the depth-wise cross-correlation is utilized. Z^{Aug} and X^{Aug} are finally combined by depth-wise cross-correlation operation, obtaining fused features R for the regression network. It can be written as:

$$R = Z^{Aug} * X^{Aug} \tag{6}$$

3.3 | Global Relevance Fusion

As discussed in the introduction, classification needs more interaction of context information and global information modeling is crucial. However, the previous cross-correlation only exploits local information to calculate similarity, which prevents the network from fully unleashing the power of context for classification. To overcome the above issue, GRF module exploits a transformer block to process and fuse feature maps. Transformers have also been shown to provide strong global reasoning. The attention module is the elementary unit in transformer block. It is used to measure the correlation between different positions of a sequence and model global information. Based on this, the GRF module consists of an encoder block and a decoder block, as shown in Figure 3. The encoder with self-attention reinforces the target features with global contextual information. The decoder contains two attention modules, one of which named self-attention is responsible for strengthening the global correlation of the search feature map. The other named cross-attention is used to mix the target features and the search features, aiming to adaptively focuses on useful information.

Encoder. The transformer encoder's primary operation is self-attention, which endeavors to mutually reinforce the features from template features $Z \in \mathbb{R}^{C\times H_Z\times W_Z}$. Where $H_Z \times W_Z$ is the spatial size, C is the dimensionality.

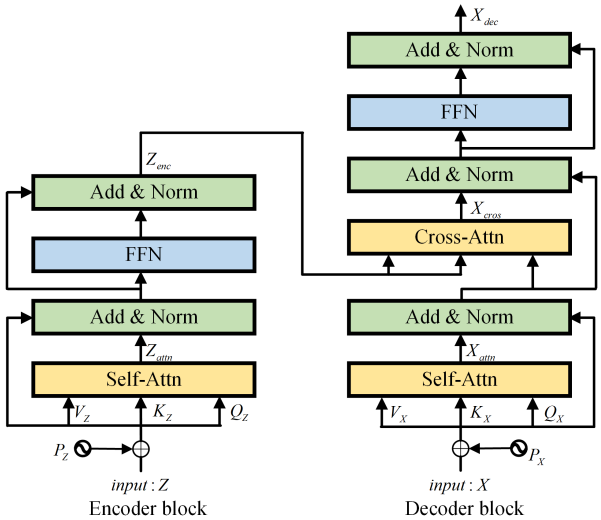

FIGURE 3 The architecture of the GRF module.

To facilitate the attention computation, we reshape Z to $Z_{flat} \in \mathbb{R}^{N_Z \times C}$, where $N_Z = H_Z \times W_Z$. Firstly, Z_{flat} is processed by a linear projection to produce Q_Z, K_Z and V_Z of the attention operation. Note that, in order to enable the network to have the perception of spatial position information, we add a spatial position encoding P_Z to the input of self-attention. The self-attention function can be expressed as:

$$\begin{aligned} Z_{attn} &= Attention(Q_Z, K_Z, V_Z) \\ &= Soft\max\left(\frac{(Q_Z + P_Z)(K_Z + P_Z)}{\sqrt{C}}\right)V_Z \end{aligned} \tag{7}$$

A residual connection is used to combine V_Z and Z_{attn}, which is followed by a normalization layer. It can be expressed by:

$$\tilde{Z} = Norm(V_Z + Z_{attn}) \tag{8}$$

Finally, the outputs of the encoder can be expressed as:

$$Z_{enc} = Norm(FFN(\tilde{Z}) + \tilde{Z}) \tag{9}$$

FFN, composed of two linear transformations, is utilized to improve the model's capacity for fitting.

Decoder. The decoder consists a self-attention and a cross attention modules. Similar to the encoder, the self-attention in decoder takes the search features $X \in \mathbb{R}^{C \times H_X \times W_X}$ as input, adaptively fuses the global features. Spectivaly, we reshape X to $X_{flat} \in \mathbb{R}^{N_X \times C}$ and produce Q_X, K_X and V_X by linear projection. In addition, we add a spatial

position encoding P_X for the search features. The self-attention can be formally denoted as:

$$\begin{aligned} X_{attn} &= Attention(Q_X, K_X, V_X) \\ &= Soft\max(\frac{(Q_X + P_X)(K_X + P_X)}{\sqrt{C}})V_X \\ \tilde{X} &= Norm(V_X + X_{attn}) \end{aligned} \tag{10}$$

Where $\tilde{X} \in \mathbb{R}^{N_X \times C}$ is the output of the self-attention. Then, we fuse X and Z by cross-attention, which contributes to the discrimination of the target from the background. It can be denoted:

$$\begin{aligned} X_{cros} &= Attention(\tilde{X}, Z_{enc}, Z_{enc}) \\ &= Soft\max(\frac{\tilde{X} \bullet Z_{enc}}{\sqrt{C}})Z_{enc}, \\ \tilde{X}_{cros} &= Norm(\tilde{X} + X_{cros}) \end{aligned} \tag{11}$$

The outputs of the decoder can be expressed as:

$$X_{dec} = Norm(FFN(\tilde{X}_{cros}) + \tilde{X}_{cros}) \tag{12}$$

Here, $X_{dec} \in \mathbb{R}^{C \times H_X \times W_X}$ is then directly fed to the classification branch.

3.4 | Tracking Framework

With the proposed SAF module and GRF module, we design a simple yet effective Siamese framework for visual tracking, named SGTrack. Our tracker is comprised of a feature extraction network, two fusion networks and a prediction network, as illustrated in Figure 2.

Feature Extraction Network. It takes an exemplar image z with size of 127 pixels and a candidate search image x with size of 255 pixels as input. A shared weight ResNet50 is used to process both inputs, and the multi-level features extracted from the last three layers are explored for layer-wise aggregation. To simplify the discussion, we will focus only on the output of the final layer, which is used to generate Z and X with size of $15 \times 15 \times 256$ and $31 \times 31 \times 256$ respectively.

These maps is then copied to (Z_{reg}, X_{reg}) and (Z_{cls}, X_{cls}) for regression and classification respectively. Z_{reg} is first centrally cropped with a size of $7 \times 7 \times 256$. SAF module takes (Z_{reg}, X_{reg}) as input and injects target semantic, generating Z_{reg}^{Aug} and X_{reg}^{Aug} with size of $7 \times 7 \times 256$ and $31 \times 31 \times 256$ respectively. They are finally combined by depth-wise cross-correlation operation, obtaining fused features R with size of $25 \times 25 \times 256$ for the regression branch. GRF module takes (Z_{cls}, X_{cls}) as input. Because the preset size of the decoder input is $25 \times 25 \times 256$, X_{cls} is spatially compressed by a convolution layer. After long-range dependencies modeling, we obtain fused features C with size of $25 \times 25 \times 256$ for the classification branch.

Prediction Network. As described above, the prediction network is comprised of a classification branch and a regression branch, each of which is implemented using several convolution layers. The classification takes C as input, generating a response map O_{cls} with size of $25 \times 25 \times 2$, which represents the probability of foreground and background. Similarly, the regression takes R as input, generating a response map O_{reg} with size of $25 \times 25 \times 4$, which represents

for d_l, d_t, d_r and d_b measuring the distance between anchor and corresponding groundtruth.

Generally speaking, we implement the asymmetrical feature fusion for Siamese tracker. In the process of back-propagation, classification and regression can better learn their respective fusion networks according to their own characteristics. Besides, the improvement of the fusion structure makes up for the simplicity of the prediction model to some extent.

3.5 | Loss

Our optimization strategy for the proposed networks involves training the regression and classification networks jointly using a combination of smooth L1 loss with normalized coordinates and cross-entropy loss. The loss function in regression is defined as

$$L_{reg} = smooth_{L_1}(x, \sigma) = \begin{cases} 0.5\sigma^2 x^2, & |x| < \frac{1}{\sigma^2} \\ |x| - \frac{1}{2\sigma^2}, & |x| \geq \frac{1}{\sigma^2} \end{cases} \tag{13}$$

where σ denotes the labels of regression offsets, x denotes the predicted offset. In classification, the loss is defined as

$$L_{cls} = -\sum p \log y + (1 - p) \log(1 - y) \tag{14}$$

where p denotes the classification labels, y denotes the predicted probability. The formula for the joint loss function is as follows,

$$loss = L_{cls} + L_{reg} \tag{15}$$

4 | EXPERIMENTS

4.1 | Implementation Details

Training. The proposed trackers employ ResNet50 as a backbone network that is initialized with pre-trained ImageNet parameters, and are trained on datasets including Youtube-BB, ImageNet VID, ImageNet DET, and COCO. The training process comprises 20 epochs, with the initial 10 epochs using a learning rate of 10^{-3} and the backbone parameters being frozen. In the remaining epochs, the backbone is unfrozen, and the tracker is trained using a decaying learning rate of 5×10^{-3} to 5×10^{-4}. The entire training process takes less than 20 hours on 4 GTX2080Ti GPUs.

Testing. We use the initial target as the template and compute feature maps once. In subsequent frames, the search region is cropped and extracted into feature maps, which is fused with feature maps of the initial frame for computing the bounding box. Utilizing a single GTX2080Ti, our tracker is capable of operating at a framerate exceeding 35 FPS.

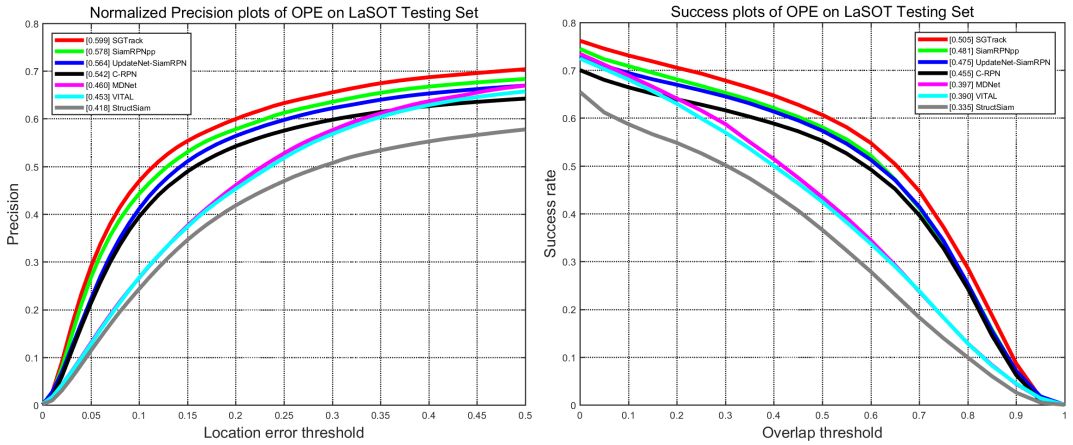

FIGURE 4 The plots on the LaSOT dataset.

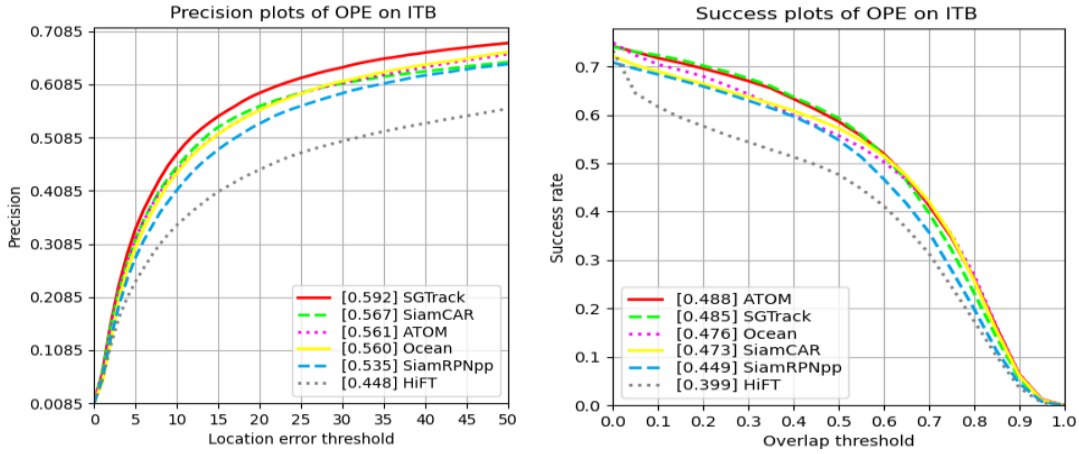

FIGURE 5 The plots on the ITB dataset.

4.2 | State-of-the-art Comparison

The proposed method is thoroughly assessed on five benchmark datasets, including LaSOT, ITB, OTB100, UAV123 and UAV20L datasets, utilizing the one pass evaluation criteria to gauge the tracking results.

LaSOT [26]. The testing dataset of LaSOT contains 280 sequences in 70 categories, each consisting of an average of 2500 frames in average. A comparison is performed between our approach and other trackers, such as UpdateNet-SiamRPN [27], C-RPN [15], MDNet [28], VITAL [29], StructSiam [30] and our baseline SiamRPN++ [5]. In both the success and precision plots, our method attains the top rank, achieving an AUC score of 50.5% and a normalized precision score of 59.9%, surpassing the baseline by 2.4%/2.1% on AUC/precision, as demonstrated in Figure 4.

ITB [31] benchmark is a recently collected dataset consisting of 180 videos and 9 scenarios. Figure 5 displays the performance of SGTrack compared with following tackers including SiamRPN++ [5], Ocean [6], ATOM [32], SiamCAR

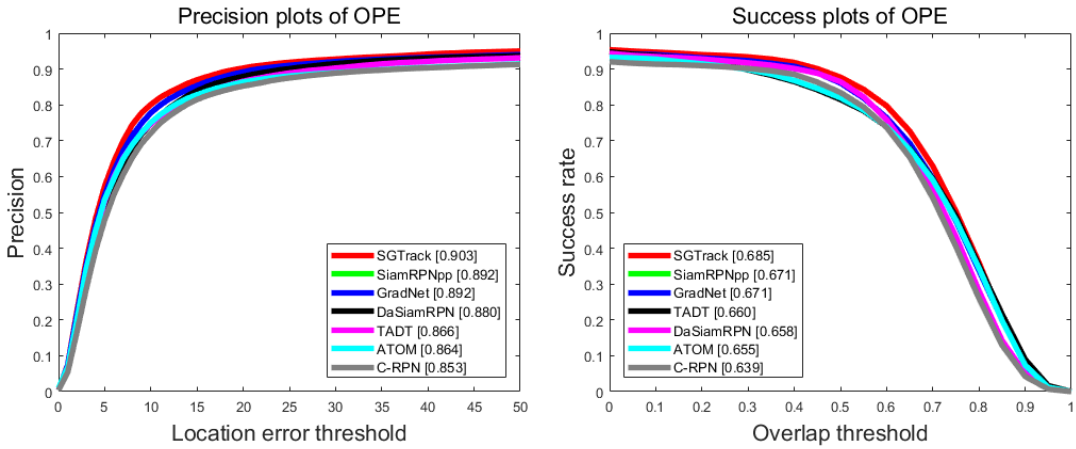

FIGURE 6 Success plot and precision plot on the OTB100 dataset.

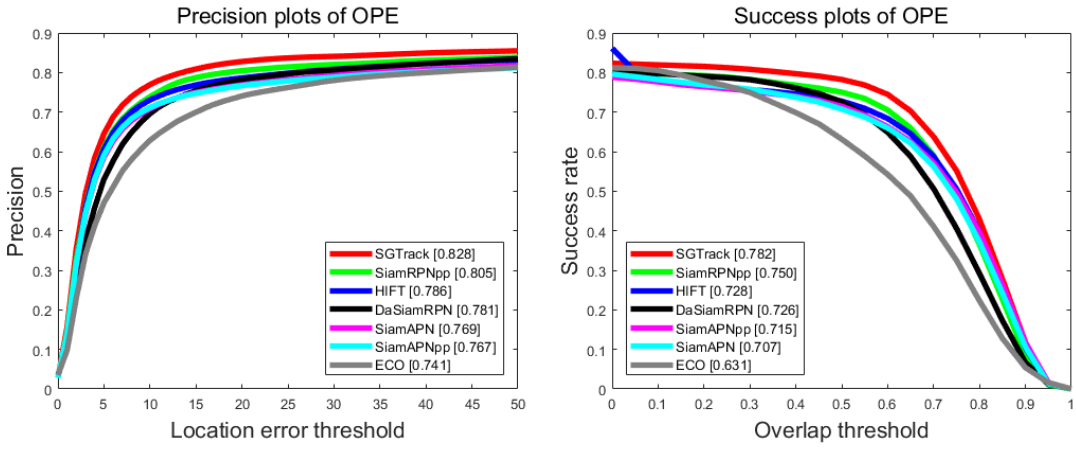

FIGURE 7 The plots on the UAV123 dataset.

[33] and HiFT [34]. Our results indicate that SGTrack's performance is second-best in terms of success, following only ATOM. However, our model attains the highest precision score, reaching 59.2%, outperforming SiamCAR, the second-best method by 2.3%. A comparison of our proposed approach with the baseline tracker SiamRPN++ reveals significant gains in precision of 5.7%. These results demonstrate that the proposed method offers high performance gains and robustness against distractors

OTB100 [35]. OTB100 is a tracking dataset comprising 100 sequences. We test our tracker on OTB100 dataset in comparison with six methods, including SiamRPN++ [5], TADT [36], GradNet [37], DaSiamRPN [14], ATOM [32] and C-RPN [15]. Figure 6 represents the results. Our tracker obtains the best performance on success compared to other trackers. Compared to the baseline tracker SiamRPN++, our approach demonstrates significant improvements of 1.1% on success.

UAV123 [38]. The benchmark dataset UAV123 comprises 123 high-quality sequences of aerial scenarios. We

TABLE 1 Tracking results on UAV20L.

UAV20L	ECO	SiamFC	DaSiam	SiamRPN++	SiamFC++	SiamCAR	ATOM	SiamAPN	SiamAPN++	HIFT	SGTrack
	[41]	[3]	[14]	[5]	[8]	[33]	[32]	[39]	[40]	[34]	
Suc.	0.427	0.402	0.465	0.564	0.533	0.544	0.554	0.539	0.560	0.566	0.587
Prec.	0.589	0.599	0.665	0.744	0.695	0.722	-	0.721	0.736	0.763	0.767

TABLE 2 Tracking results under different attributes on LaSOT.

	IV	POC	DEF	MB	CM	ROT	BC	VC	SV	FOC	FM	OV	LR	ARC
StructSiam	0.366	0.310	0.361	0.309	0.337	0.319	0.320	0.246	0.331	0.241	0.215	0.278	0.258	0.313
VITAL	0.403	0.361	0.384	0.363	0.397	0.371	0.365	0.339	0.385	0.301	0.246	0.304	0.309	0.358
MDNet	0.407	0.370	0.391	0.376	0.416	0.379	0.374	0.358	0.392	0.305	0.260	0.330	0.317	0.366
C-RPN	0.487	0.432	0.479	0.413	0.482	0.438	0.409	0.405	0.452	0.348	0.29	0.365	0.355	0.435
UpdataNet-SiamRPN	0.504	0.452	0.497	0.439	0.500	0.456	0.425	0.454	0.474	0.373	0.302	0.398	0.385	0.456
SiamRPN++	0.537	0.450	0.506	0.438	0.496	0.469	0.434	0.435	0.479	0.362	0.336	0.397	0.385	0.460
SGTrack	0.531	0.473	0.529	0.466	0.524	0.498	0.445	0.482	0.504	0.392	0.350	0.440	0.414	0.481

compare with six methods including SiamRPN++ [5], HIFT [34], DaSiamRPN [14], SiamAPN [39], SiamAPN++ [40] and ECO [41]. Results are presented in precision and success plots, depicted in Figure 7. Our tracker achieves the best results on both plots. Compared to the baseline, we obtained an increase of 3.2% and 2.3% in success and precision respectively.

UAV20L. The UAV20L consists of 20 long videos, each with an average of 2934 frames and in total of over 58K frames. It is utilized to evaluate the performance in long-term aerial tracking scenarios. See Table 1, we compare with ten trackers. Our method exhibits superior performance, attaining the highest success (58.7%) and precision (76.7%). Futhermore compared to the baselien, we observe a notable improvement of 2.3% and 2.3% in success and precision respectively, confirming the superiority of our method.

4.3 | Attribute Comparison

We further investigate the results under different situations and challenges on the LaSOT. Table 2 presents the attribute based evaluation for overlap success rates. It shows that the proposed method is more effective on most attributes. In addition, our method surpasses the baseline SiamRPN++ on 13 attributes in the whole 14 categories, demonstrating its overall performance gains. Especially, the impressive efficacy on attributes viewpoint change (+4.7%), full occlusion (+3.0%), and out-of-view (+4.3%) clearly demonstrates the power of the proposed tracker. In these case the target is easily lost, and we infer that SGTrack can effectively re-detect the object by exploiting target's semantic information.

4.4 | Ablation Experiment

Our method contains two components: SAF (semantic augmentation fusion) and GRF (global relevance fusion) components. We implement several variants and evaluate their performance on the ITB dataset. The corresponding results are presented in Table 1, which highlights distinct tendencies of classification and regression towards various types of information. In addition, the enhancement of semantic information leads to a more remarkable improvement in the

TABLE 3 Ablation analysis on ITB testing set.

Classification	Regression	Precision
×	×	0.535
GRF	×	0.555
×	GRF	0.543
SAF	×	0.537
×	SAF	0.580
GRF	SAF	0.592

regression branch. We believe that the regression branch has more potential to improve performance.

4.5 | Visual Comparison

Figure 8 illustrates our qualitatively validate on five sequences of LaSOT with different challenging attributes: *lion* – 5, *rabbit* – 10, *sheep* – 9, *tank* – 14 and *chameleon* – 20.The compared methods include UpdateNet-SiamRPN, C-RPN, MDNet, VITAL, StructSiam and the baseline SiamRPN++.

lion – 5. The main challenges of sequence *lion* – 5 are full occlusion, background clutter and deformation. Occlusion starts at frame #379, and trackers start to drift. When full occlusion occurs at frame #518, trackers lose the target. The target reappears at frame #1155 with background clutter and deformation. Most trackers thus can't recognize

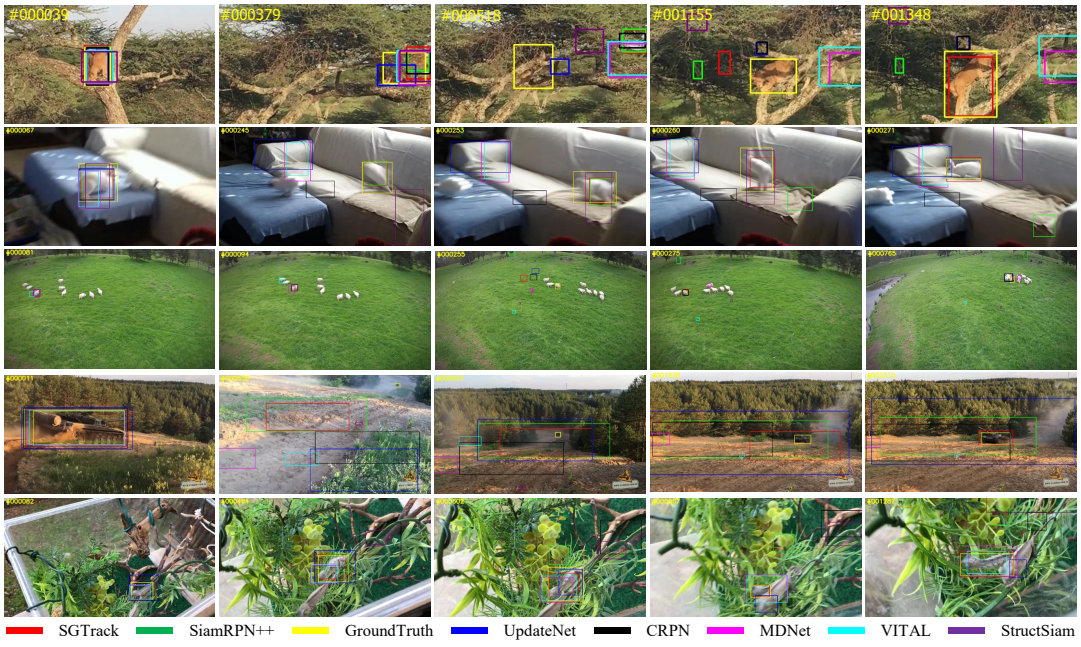

FIGURE 8 We conducted a visual comparison of our tracker with the following trackers: SiamRPN++, UpdateNet-SiamRPN, C-RPN, MDNet, VITAL and StructSiam.

the target, while SGTrack can re-track the target due to its powerful semantic appearance correlation ability.

rabbit – 10. Fast motion easily leads to model drift. When running to frame #253, only SGTrack, SiamRPN++ and StructSiam can locate the target. Then, the background is similar to the color of the target, causing SiamRPN++ and StructSiam drift and eventual tracking failure. SGTrack can locate the target continuously because of its high robustness against distractors.

sheep – 9. The sharp rotation of the camera view causes all trackers to lose the target at frame #255. When the view is restored, SGTrack, C-RPN and UpdateNet-SiamRPN re-locate the target. At frame #766, C-RPN and UpdateNet-SiamRPN drift again due to similar objects, while SGTrack can keep tracking.

tank – 14. In this sequence, the target size changes dramatically, leading to all trackers tracking failure. When the target size is recovering, our tracker can immediately re-detect the target and adapt to the target size at frame #2310.

chameleon – 20. The main challenge of this sequence is that background clutter is extremely serious. At frame #464, trackers except SGTrack suffer from model drift or tracking failure because the target color is similar to the tree. When the similar target occurs, the predictions of many trackers are inaccurate while SGTrack predicts the correct position and size.

Generally speaking, these sequences show our tracker can re-detect the target and against distractors, which proves our methond's effectiveness.

5 | CONCLUSION

The current research introduces a novel tracking architecture, demonstrating the potential of exploiting the distinctive characteristics of classification and regression to optimize feature pre-processing modules and fusion strategies. We exploit a SAF module to inject the target semantic into the regression branch, which relates the semantic and appearance of the target and guides the regression network for more accurate estimation. Besides, we use a GRF module to capture global information model long-range dependencies for effectively mixing the target and search area information, providing high robustness against distractor objects for classification. Upon verification, the proposed method has demonstrated enhanced tracking performance while still maintaining a significant speed.

references

[1] Wu Y, Lim J, Yang MH. Online Object Tracking: A Benchmark. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2013. p. 2411–2418.

[2] Tao R, Gavves E, Smeulders AWM. Siamese Instance Search for Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2016. p. 1420–1429.

[3] Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PHS. Fully-Convolutional Siamese Networks for Object Tracking. In: Proc. Eur. Conf. Comput. Vis. (ECCV); 2016.p. 850–865.

[4] Li B, Yan J, Wu W, Zhu Z, Hu X. High Performance Visual Tracking with Siamese Region Proposal Network. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2018. p. 8971–8980.

[5] Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2019. p. 4282–4291.

[6] Zhang Z, Peng H, Fu J, Li B, Hu W. Ocean: Object-Aware Anchor-Free Tracking. In: Proc. Eur. Conf. Comput. Vis. (ECCV); 2020.p. 771–787.

[7] Chen Z, Zhong B, Li G, Zhang S, Ji R. Siamese Box Adaptive Network for Visual Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2020. p. 6668–6677.

[8] Xu Y, Wang Z, Li Z, Ye Y, Yu G. SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines. In: Proc. Conf. AAAI. Artif. Intell.; 2020. p. 12549–12556.

[9] Li D, Porikli F, Wen G, Kuai Y. When Correlation Filters Meet Siamese Networks for Real-Time Complementary Tracking. IEEE Trans Circuits Syst Video Technol 2020 feb;30(2):509–519.

[10] Pi Z, Shao Y, Gao C, Sang N. Instance-Based Feature Pyramid for Visual Object Tracking. IEEE Trans Circuits Syst Video Technol 2022 jun;32(6):3774–3787.

[11] Jiang M, Zhao Y, Kong J. Mutual Learning and Feature Fusion Siamese Networks for Visual Object Tracking. IEEE Trans Circuits Syst Video Technol 2021 aug;31(8):3154–3167.

[12] Fan J, Song H, Zhang K, Yang K, Liu Q. Feature Alignment and Aggregation Siamese Networks for Fast Visual Tracking. IEEE Trans Circuits Syst Video Technol 2021 apr;31(4):1296–1307.

[13] Yao R, Lin G, Shen C, Zhang Y, Shi Q. Semantics-Aware Visual Object Tracking. IEEE Trans Circuits Syst Video Technol 2019 jun;29(6):1687–1700.

[14] Zhu Z, Wang Q, Li B, Wu W, Yan J, Hu W. Distractor-Aware Siamese Networks for Visual Object Tracking. In: Proc. Eur. Conf. Comput. Vis. (ECCV); 2018.p. 103–119.

[15] Fan H, Ling H. Siamese Cascaded Region Proposal Networks for Real-Time Visual Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2019. p. 7952–7961.

[16] Zhang Z, Peng H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2019. p. 4591–4600.

[17] Hu J, Shen L, Albanie S, Sun G, Wu E. Squeeze-and-Excitation Networks. IEEE Trans Pattern Anal Mach Intell 2020 aug;42(8):2011–2023.

[18] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Adv Neural Inf. Process. Syst.; 2017. p. 4–9.

[19] Wang X, Girshick R, Gupta A, He K. Non-local Neural Networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2018. p. 7794–7803.

[20] Choi J, Chang HJ, Yun S, Fischer T, Demiris Y, Choi JY. Attentional Correlation Filter Network for Adaptive Visual Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2017. p. 4807–4816.

[21] He A, Luo C, Tian X, Zeng W. A Twofold Siamese Network for Real-Time Object Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2018. p. 4834–4843.

[22] Wang Q, Teng Z, Xing J, Gao J, Hu W, Maybank S. Learning Attentions: Residual Attentional Siamese Network for High Performance Online Visual Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2018. p. 4854–4863.

[23] Shen J, Tang X, Dong X, Shao L. Visual Object Tracking by Hierarchical Attention Siamese Network. IEEE Trans Cybern 2020 jul;50(7):3068–3080.

[24] Gao P, Zhang Q, Wang F, Xiao L, Fujita H, Zhang Y. Learning reinforced attentional representation for end-to-end visual tracking. Inf Sci 2020 may;517:52–67.

[25] Tan H, Zhang X, Zhang Z, Lan L, Zhang W, Luo Z. Nocal-Siam: Refining Visual Features and Response With Advanced Non-Local Blocks for Real-Time Siamese Tracking. IEEE Trans Image Process 2021;30:2656–2668.

[26] Fan H, Lin L, Yang F, Chu P, Deng G, Yu S, et al. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2019. p. 5374–5383.

[27] Zhang L, Gonzalez-Garcia A, Weijer JVD, Danelljan M, Khan FS. Learning the Model Update for Siamese Trackers. In: Proc. IEEE Int. Conf. Comput. Vis. (ICCV); 2019. p. 4010–4019.

[28] Nam H, Han B. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2016. p. 4293–4302.

[29] Song Y, Ma C, Wu X, Gong L, Bao L, Zuo W, et al. VITAL: VIsual Tracking via Adversarial Learning. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2018. p. 8990–8999.

[30] Zhang Y, Wang L, Qi J, Wang D, Feng M, Lu H. Structured Siamese Network for Real-Time Visual Tracking. In: Proc. Eur. Conf. Comput. Vis. (ECCV); 2018.p. 355–370.

[31] Li X, Liu Q, Pei W, Shen Q, Wang Y, Lu H, et al. An Informative Tracking Benchmark. In: arXiv preprint arXiv:2112.06467; 2021. .

[32] Danelljan M, Bhat G, Khan FS, Felsberg M. ATOM: Accurate Tracking by Overlap Maximization. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2019. p. 4660–4669.

[33] Guo D, Wang J, Cui Y, Wang Z, Chen S. SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2020. p. 6269–6277.

[34] Cao Z, Fu C, Ye J, Li B, Li Y. HiFT: Hierarchical Feature Transformer for Aerial Tracking. In: Proc. IEEE Int. Conf. Comput. Vis. (ICCV); 2021. p. 15457–15466.

[35] Wu Y, Lim J, Yang MH. Object Tracking Benchmark. IEEE Trans Pattern Anal Mach Intell 2015 sep;37(9):1834–1848.

[36] Li X, Ma C, Wu B, He Z, Yang MH. Target-Aware Deep Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2019. p. 1369–1378.

[37] Li P, Chen B, Ouyang W, Wang D, Yang X, Lu H. GradNet: Gradient-Guided Network for Visual Object Tracking. In: Proc. IEEE Int. Conf. Comput. Vis. (ICCV); 2019. p. 6162–6171.

[38] Mueller M, Smith N, Ghanem B. A Benchmark and Simulator for UAV Tracking. In: Proc. Eur. Conf. Comput. Vis. (ECCV); 2016.p. 445–461.

[39] Fu C, Cao Z, Li Y, Ye J, Feng C. Siamese Anchor Proposal Network for High-Speed Aerial Tracking. In: IEEE Int. Conf. Robot. Autom. (ICRA); 2021. p. 510–516.

[40] Cao Z, Fu C, Ye J, Li B, Li Y. SiamAPN++: Siamese Attentional Aggregation Network for Real-Time UAV Tracking. In: Proc. IEEE Int. Conf. Intell. Rob. Syst. (IROS); 2021. p. 3086–3092.

[41] Danelljan M, Bhat G, Khan FS, Felsberg M. ECO: Efficient Convolution Operators for Tracking. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR); 2017. p. 6638–6646.