

Article

SiamOT: An Improved Siamese Network with Online Training for Visual Tracking

Xiaomei Gong, Yuxin Zhou and Yi Zhang * 

Department of Computer Science, Sichuan University, Chengdu 610017, China

* Correspondence: yi.zhang@scu.edu.cn

Abstract: As a prevailing solution for visual tracking, Siamese networks manifest high performance via convolution neural networks and weight-sharing schemes. Most existing Siamese networks have adopted various offline training strategies to realize precise tracking by comparing the extracted target features with template features. However, their performances may degrade when dealing with unknown targets. The tracker is unable to learn background information through offline training, and it is susceptible to background interference, which finally leads to tracking failure. In this paper, we propose a twin-branch architecture (dubbed SiamOT) to mitigate the above problem in existing Siamese networks, wherein one branch is a classical Siamese network, and the other branch is an online training branch. Especially, the proposed online branch utilizes feature fusion and attention mechanism, which is able to capture and update both the target and the background information so as to refine the description of the target. Extensive experiments have been carried out on three mainstream benchmarks, along with an ablation study, to validate the effectiveness of SiamOT. It turns out that SiamOT achieves superior performance with stronger target discrimination abilities.

Keywords: visual tracking; Siamese networks; online training



Citation: Gong, X.; Zhou, Y.; Zhang, Y. SiamOT: An Improved Siamese Network with Online Training for Visual Tracking. *Sensors* **2022**, *22*, 6597. <https://doi.org/10.3390/s22176597>

Academic Editor: Radu Danescu

Received: 6 July 2022

Accepted: 9 August 2022

Published: 1 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Visual tracking is one of the most fundamental and active topics in the computer vision domain. Given the initial state (including the position and size of the bounding box) of an object in the first frame with virtually no prior knowledge, the tracker needs to continuously track the state of the object in the subsequent frames. In early years, template-based methods and correlation filters were the mainstream methods. For instance, the MOSSE [1] filter was a pioneering work in visual tracking using a correlation filter. In the next few years, with the growing popularity of deep learning, Siamese-based trackers sprung up with appealing performances. Albeit effective, they are facing two problems: First of all, most of them adopted offline training schemes, which mainly focused on the target features but overlook the background characteristics. The background information is also of vital importance to tracking tasks, especially the target surroundings. Secondly, an offline training scheme trains the network using image pairs, which could neither adapt to target changes nor recognize the unknown targets (those do not appear in the training sets). To realize end-to-end training, DiMP [2] announced a dedicated optimization and loss function, which fully exploited both target and background appearance information to predict target model within a few iterations. GradNet [3] is a novel gradient-guided network that updated the template in the Siamese network through feed-forward and backward operations. However, its shallow network did not well resolve the problem of background changes, which resulted in low tracking performance.

To remedy the above defects of Siamese networks, we propose integrating online training into Siamese networks in this paper. Compared with traditional Siamese networks, the online training branch could easily recognize the unknown targets, which weakens the influence of other disturbing factors. Combining the abundant target features extracted by

the Siamese network with the online learned information would undoubtedly improve the tracking performance. In addition, we also employ feature fusion and attention mechanisms in our proposed online branch to make our tracker focus on more important target features.

A sequence of tracking results is illustrated in Figure 1 between SiamOT and other popular methods (DaSiamRPN [4], ECO [5], GradNet [3], and SiamFC [6]). The black boxes are ground truth, and are manually annotated in each frame. The yellow boxes indicate our tracking results. Initially, all trackers target the wild goose. However, starting from the second frame, all other trackers either drift to other, similar geese, or point to the cloud clusters; only SiamOT always keeps track of the original goose.



Figure 1. Comparative results among different trackers.

In a nutshell, the main contributions to this paper can be summarized as follows:

1. We develop an online training module for a Siamese-based tracker, which aims to capture and update both the target and background changes. It turns out that the proposed module improves the target recognition ability of the tracker.
2. We investigate several feature fusion methods and propose a weighted fusion scheme, which integrates both deep and shallow features extracted by different layers of the backbone.
3. We devise an improved attention module and insert it into our tracking framework to make the tracker pay attention to more important channels so as to improve tracking efficiency.
4. Extensive experiments have been conducted on three mainstream benchmarks, along with ablation studies, to verify our design. SiamOT exhibits favorable performances.

The rest of the paper is organized as follows: Related works are discussed in Section 2. The architecture of our network is described in Section 3. The experimental results with ablation studies are shown in Section 4 with thorough analysis. A final conclusion is drawn in Section 6.

2. Related Works

Normally, a generic tracking problem is decomposed into two steps, namely target classification and state estimation. The former step detects the target of interest in each frame, while the latter step regresses the properties of the target in the form of a bounding box (including central position and box size) that best describes the target in the current frame. Then, the tracker can associate the target across frames to form a trajectory. In view of the essence of tracking problems, the Siamese network became a suitable solution for visual tracking due to its two-branch structure, in which one branch is committed to classification and the other branch is used for regression. As a result, Siamese networks began to predominate the visual tracking field with astonishing advancement.

2.1. Siamese Networks

Considering the strong feature representation abilities of convolution neural networks (CNNs), SiamFC [6] applied the Siamese network to visual tracking for the first time using an offline training scheme. It generated the target features for both the first and the current frames based on Siamese structure, calculated their correlation to yield a response map, and finally located the target based on the position of its peak value on the response map. SPM-Tracker [7] elucidated a series-parallel matching algorithm to strengthen the robustness

of the tracker on the basis of SiamFC. To improve the localization accuracy, SiamRPN [8] advocated a two-branch architecture, where one branch was responsible for classification (to distinguish target from the background) and the other branch was dedicated to a regression task (to estimate the position of target bounding box). In particular, SiamRPN [9] introduced the Region Proposal Network (RPN) for regression tasks. SiamMask [10] appended detection head to SiamRPN to create semantic segmentation results for the target. Zhu et al. [4] pointed out that the major drawback of current Siamese networks is the imbalanced distribution of training data, making the learned features less discriminative. They designed a distractor-aware module to perform incremental learning so as to facilitate target classification. Inspired by SiamRPN, Fan et al. [11] elaborated a multi-stage visual tracking framework to improve tracking accuracy and robustness, which was comprised of a sequence of RPNs cascading from high-level to low-level layers in a Siamese network.

SiamRPN++ [12] was an improved version of SiamRPN, which analyzed the reason why deeper networks presented worse results than shallow networks. It then adopted a revised ResNet-50 as the backbone by replacing part of the convolution blocks with dilated convolutions. It also utilized anchor-based mechanism for object detection. However, an anchor-based mechanism needs prior knowledge about the ratio of the bounding box. Therefore, the tracker could not locate a newly appeared target that was not consistent with the prior knowledge. Zhang et al. [13] replaced the original AlexNet [14] backbone with a deeper and wider convolution network to improve robustness and accuracy. It also designed new cropping-inside residual units (CIR) to eliminate the negative impact of network padding. To overcome the problem of data drift, Tang et al. [15] provided a template adjustment module to adapt to the target appearance variation in a long-term sequence.

To overcome the inherent limitations of anchor-based mechanisms coping with unknown targets, SiamPCF [16] located targets in an adaptive manner. Instead of using the bounding box directly, SiamPCF transformed the points on the target to a bounding box to describe the target more accurately. SiamBAN [17] was an anchor-free based tracker, which announced an elliptical-shaped labeling method to locate the target. AF2S [18] replaced the anchor-based RPN with a two-stage tracker based on SiamFC. Zhang et al. [19] introduced a feature alignment module to learn object-aware features within the predicted bounding boxes. Peng et al. [20] regressed the attributes of the bounding box without additional classification or regional proposals, and tuned the stride and receptive field for the network. Compared with the abovementioned methods, Held et al. [21] realized a high-speed tracking network (100 fps) via simple feed-forward network without online training. Despite its high speed, its accuracy was less satisfactory, since it derived the target position simply based on previous frame.

The core idea of Siamese networks is the calculation of similarity between the template patch and the search patch. However, most existing Siamese networks adopted offline training schemes, which lacked the learning and updating of target changes during tracking. Moreover, some Siamese networks excel at identifying the target from the background, but they are easily disturbed by similar targets or other interferences.

2.2. Template Update Algorithms for Visual Tracking

ATOM [22] was a novel and high-speed tracker, which incorporated dedicated target estimation and classification components. It computed the overlapped region (IoU: intersection over union) between the predicted and ground truth boxes and maximized the IoU to locate the target using a gradient descent method. UpdateNet [23] replaced the handcrafted update function with a lightweight auto-learning module. An optimal template for the next frame was estimated based on a combination of the initial template, the accumulated template, and the current frame. GradNet [3] was a novel gradient-guided network used to solve the over-fitting problem, which exploited the discriminative information in gradients and updated the template through feed-forward and backward operations. However, it suffered the same limitations as other methods of short memory of

the past frames. For this reason, LSTM modules were implanted to store historical information [24–26], but they were computationally heavy. MDNet [27] encompassed shared layers and multiple branches of domain-specific layers to identify the target in each domain. However, it failed to meet real-time demand. MBMD [28] developed a long-term tracking framework based on a deep regression and verification module to generate a series of candidates and to predict their similarity scores based on object-aware feature fusion and RPN. Yao et al. [29] investigated the joint learning of deep representation and model adaptation, and truncated the alternating direction method of multipliers (ADMM). Choi et al. [30] put forth a template selection strategy using deep reinforcement learning methods.

A variety of template update strategies have been developed by the above-mentioned methods, with proven improvements. Nevertheless, simple update mechanisms could not handle unpredictable target changes or complex scenes, while more complicated algorithms would inevitably increase the computational cost.

2.3. Emerging Methods in Visual Tracking

Recently, meta-learning has been widely applied to visual tracking. Park et al. [31] developed an offline meta-learning-based method to adjust the initial deep networks used in online adaptation-based tracking. On the other hand, motivated by the great success of transformer in natural language processing (NLP), researchers attempted to migrate it to a computer vision domain to implement downstream vision tasks. Aiming to solve the insufficient global information and weak feature extraction ability of existing trackers, Hu et al. [32] advised using a novel tracking architecture that combined feature enhancement and template update. Yan et al. [33] came up with an encoder–decoder transformer to estimate the positions of target corners. However, some of the transformer architectures relied on high-end equipment (e.g., $8 \times$ V100 cards are required to run [33]). Except for a few commercial institutions, other units cannot afford such an expensive platform to replicate the results.

Considering that Siamese networks are still one of the dominant methods in visual tracking, in this paper, we mainly focus on improving the tracking performance of Siamese-based networks using an online training scheme. Especially, we implant an online training branch to enhance the adaptation ability of target changes. Our network structure will be described in detail in Section 3 below.

3. Siamese Networks with Online Training Module

The overall structure of SiamOT is illustrated in Figure 2 below. We use ResNet-50 as our backbone. The yellow and the green blocks represent classification (cls) and regression (reg) branches, respectively, and the blue block denotes the online training (OT) branch.

3.1. Problem Formulation

For a typical Siamese network, the score maps of classification and regression branches are written as:

$$\begin{cases} s(z, x)^{cls} = \sum_{i=3}^5 \alpha_i (\varphi_i(z) * \varphi_i(x)) \\ s(z, x)^{reg} = \sum_{i=3}^5 \beta_i (\varphi_i(z) * \varphi_i(x)) \end{cases}. \quad (1)$$

Here, z and x represent the cropped patches in the first frame and the current frame. α_i and β_i are the weights for classification and regression branches, respectively. φ_i is the feature extractor. $*$ refers to the convolution operation.

The purpose of the proposed online training module is to capture and update the target changes. The score map of the online training module is calculated as:

$$s^{online} = s_C(Att(f(x)); \theta) \quad (2)$$

Here, $f(x)$ indicates the extracted feature of x . Att represents the attention mechanism. θ is the set of weights of online training module; s_C refers to an online training function. As shown in Figure 2, s^{online} is fused with the output of classification branch ($s(z, x)^{cls}$) to better distinguish the target from the background. The details of the entire network will be described in Section 3.2 below.

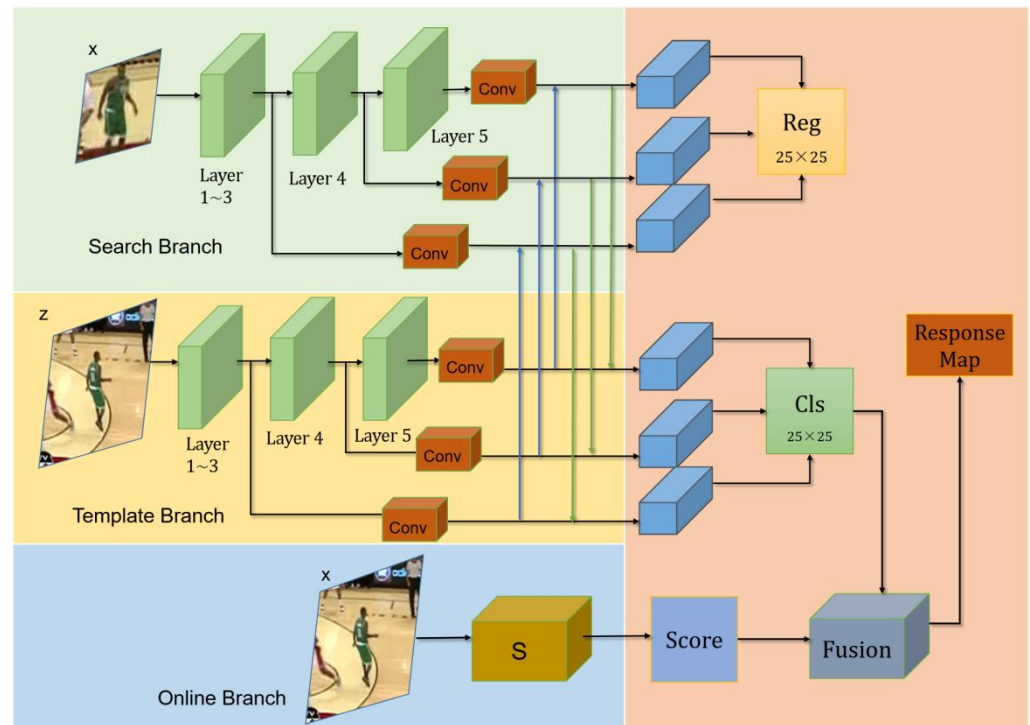


Figure 2. The overall structure of SiamOT.

3.2. Online Training Module (OT)

The online training module is illustrated in Figure 3. It consists of a feature fusion sub-module (FF), a feature attention sub-module (FA), and a training sub-module (s). The features from the third, the fourth, and the fifth layer of the backbone are first processed by convolution blocks, and are then fused by FF. Then, the important features are filtered by FA, and are finally fed to s to yield the final score map s^{online} . During training, x is augmented by our OT branch to create a training set, and the weights of s are also updated at the same time. The OT branch is explained in detail in Section 3.2.4.

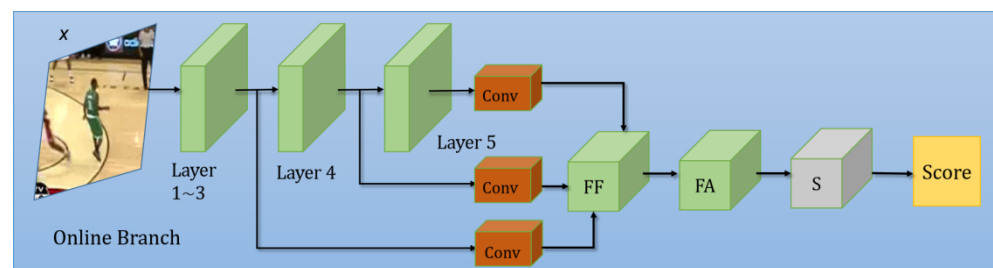


Figure 3. The online training (OT) branch of our network.

3.2.1. Feature Fusion (FF)

The feature that is fed to the training sub-module is critical for online training. Some of the previous methods (published after SiamRPN++) employed ResNet-50 as the backbone in a weight-sharing manner. Other methods [20] directly utilized the deep features from the fifth layer. Normally, the deep features contain more semantic information, and the

shallow features carry spatial details, which are all beneficial for tracking. Considering this, we investigate several feature fusion methods, including simple addition (3), multiply (4), and concatenation (5) as follows:

$$f(x)_{add} = \sum_{i=3}^5 \varphi_i(x) \quad (3)$$

$$f(x)_{Multiply} = \prod_{i=3}^5 \varphi_i(x) \quad (4)$$

$$f(x)_{Concat} = \text{concat}(\varphi_3(x), \varphi_4(x), \varphi_5(x)). \quad (5)$$

Here, $\varphi_3(x)$, $\varphi_4(x)$ and $\varphi_5(x)$ denote the extracted feature after the third, the fourth, and the fifth layer of backbone (shown in Figure 3 above). *concat* denotes the concatenation operation.

However, the above three fusion methods yield less satisfactory results. Thus, we try the following weighted fusion method:

$$f(x)_{wf} = \sum_{i=3}^5 \alpha_i \varphi_i(x) \quad (6)$$

where α_i is the weight for each extracted feature. It turns out that the weighted fusion method produces the best results. A comparison of the above methods (3)–(6) is provided in the ablation study in Section 5.1.1.

3.2.2. Attention Mechanism

SiamRPN++ [12] visualizes the features of different channels to prove that only part of the channels contribute to the final score map. Inspired by [12,34], we add an attention module into our tracking framework to focus more on important channels. Our proposed attention assisted network architecture is shown in Figure 4, in which input $f(x)$ is a fused feature calculated by (6) above. Instead of directly using the structure in [34], we insert channel attention and spatial attention modules into the original structure to calculate the corresponding attentions in the feature map so as to pay more attention to the important channel and spatial location.

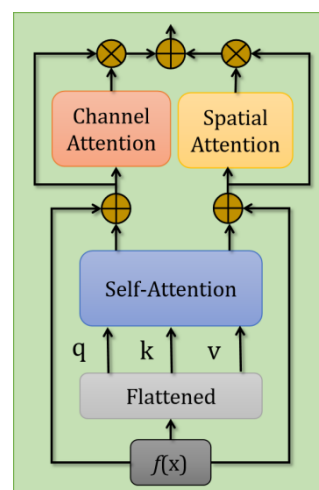


Figure 4. The architecture of the proposed attention module in SiamOT.

In order to strengthen the connection between each $1 \times 1 \times 256$ feature of $f(x)$, the fused feature is flattened into q, k, v (query, key, value) components and are enhanced as:

$$Att_s = \text{softmax}(qk^T \delta) v + f(x) \quad (7)$$

Here, δ is a weight with the same dimension as $f(x)$. Att_s is the enhanced feature via attention mechanism (with dimension $\text{channel} \times w \times h$). First of all, we flatten $f(x)$ ($h \times w \times 256$) into $hw \times 256$. Meanwhile, q is $hw \times 256$ and k^T is $256 \times hw$, so qk^T calculates the relation between each $1 \times 1 \times 256$ dimensional feature and the remaining $h \times w - 1$ features via inner product multiplication. Then, *softmax* function is used to normalize the relation of each row. Finally, the normalized results are added by the original $f(x)$ to yield rich target features (such as a residual structure).

The results of the self-attention module are fed into channel and spatial attention modules, respectively, for further feature enhancement in channel and spatial dimensions. To be more specific, the following equation is devised for the channel attention module:

$$W_{channel} = \text{Sigmoid}(\text{Relu}(\text{Avgpool}(Att_s)w_{c1})w_{c2}) \quad (8)$$

Here, *Avgpool* refers to average pooling operation; w_{c1} and w_{c2} are the corresponding weights. Firstly, we perform an *Avgpool* on the enhanced features Att_s , followed by *Relu* and *Sigmoid* activation functions to produce the final weights for the channel attention module $W_{channel}$.

Similarly, for spatial attention module, we generate the weights as:

$$W_{spatial} = \text{Sigmoid}(Att_m) \quad (9)$$

Here, Att_m (with dimension $1 \times w \times h$) is the mean value of Att_s along the channel dimension (after the mean operation, the channel size becomes one). Then, the features processed by spatial and channel attention modules are expressed as:

$$Att_{spatial} = Att_s \otimes W_{spatial} \quad (10)$$

$$Att_{channel} = Att_s \otimes W_{channel} \quad (11)$$

$\otimes$ denotes element-wise multiplication. Finally, a fusion operation is performed as follows:

$$Att = Att_{channel} \oplus Att_{spatial} \quad (12)$$

$\oplus$ denotes element-wise addition.

3.2.3. Online Learning Function

Danelljan et al. [2] declared that classical Siamese networks could only learn target features from image pairs through offline training. In contrast, online trained networks could learn both the target and the background information. In this light, we append the online training branch to our architecture. After a series of feature fusion and augmentation operations, we apply a learning function (s in Figure 3) to calculate the score map according to (2), so as to distinguish target from other distractions. The loss of the proposed learning function is calculated as:

$$L(s) = \|s^{online} - y_g\|^2 \quad (13)$$

Here, $L(s)$ is the loss value of online training, s^{online} represents the score map in (2), and y_g is the ground truth value. We adopt Gaussian distribution to model the target position, using y_g as the center.

Compared with offline training, the online training scheme captures richer target features. However, it has sparse training samples; thus, we utilize historical frames as the training sample with data augmentation techniques.

3.2.4. The Fusion of Siamese Network with Online Training Branch

As described earlier, Siamese networks and online training schemes have their own advantages. For instance, the former could be trained by a large amount of image pairs to ensure high recognition accuracy. The latter updates the target changes. Based on this, we

combine them together to compute a more accurate score map. The most straightforward way to combine them is by simply adding them together:

$$s(z, x)^{final} = s(z, x)^{cls} + s^{online} \quad (14)$$

However, the result is less satisfactory. Instead, it turns out that a weighted fusion demonstrates much better results:

$$s(z, x)^{final} = (1 - \eta) \times s(z, x)^{cls} + \eta \times s^{online} \quad (15)$$

Here, η is set to 0.8. An ablation study is conducted in Section 5.1.3 to verify this empirical value.

3.2.5. Online Tracking Algorithm Flow

In this section, we will explain the processing flow of our network. Given the first frame, the input z is generated based on the target position. Then, a data augmentation method [35] is applied to the first frame to yield the training set T , which is used for initial online training. During tracking, we send x (generated by the test frame) to the two branches to produce two score maps, which are fused to obtain the target center. Finally, the target bounding box is created by the regression branch. The entire online tracking flow is written in Algorithm 1. The parameters involved in online training and template updating will be discussed in Section 4 below.

Algorithm 1 Flow of Online Tracking

Input: The 1st frame of the video sequence I_0 ; The initial target bounding box b_0

Output: The current target bounding box b_t

- 1: Create cropped patch z based on b_0 from first frame I_0 . Create the training set T based on I_0 , which is used to online train the network s .
 - 2: **for** $t = 1, \dots, N$ **do**
 - 3: Create search patch x based on I_t
 - 4: Obtain $s^{cls}(z, x)$ using Equation (1)
 - 5: Obtain $f(x)$ using Equation (6)
 - 6: Obtain Att_s using Equation (7)
 - 7: Obtain Att on the basis of Att_s using Equations (8)–(12)
 - 8: Calculate score map s^{online} using Equation (2)
 - 9: Calculate s^{final} using Equation (14)
 - 10: Obtain the target bounding box in current frame based on s^{final} and $s^{reg}(z, x)$
 - 11: **if** $(t \bmod C) = 0$ **then**
 - 12: Obtain the cropped patch $\varphi(z)$ based on b_t
 - 13: Replace the template of Siamese network by current frame $\varphi(z_0) = \varphi(z)$
 - 14: **end if**
 - 15: **end for**
-

4. Experimental Details

The following experiments are built on the Pytorch framework with Python. The detailed experimental configurations are as follows (Table 1):

Table 1. Experimental environment.

Hardware	GPU	Nvidia RTX 3090
	CPU	Intel® Core™ i7-9700 K CPU @ 3.60 GHz × 8
Software	OS	Ubuntu 20.04
	Framework	Pytorch 1.7.0
	Language	Python

The initial weights of our network are set by referring to SiamBAN. Since we adopt online training scheme, the traditional gradient update algorithms are not involved and discussed here. We employ translation, flip, blurring, and rotation for data augmentation to produce 30 training samples with dimensions $31 \times 31 \times 256$. Then, the Gaussian score map is generated as the ground-truth label, in which the real target center is treated as its center.

After the generation of training samples and ground-truth values, we perform 10 iterations of gradient descents to initialize the optimizer. After initialization, all training samples are sent to attention module for another 60 iterations of training. In the following tracking process, feature fusion and enhancement are carried out for each input patch x . s^{online} is calculated through Figure 3 to fuse the final score map via (15). As suggested by [5,22,34] The update interval (C in Table 1) is set to five.

5. Experiment

In this section, we conduct comparative experiments on three popular datasets, including OTB100 [36], LaSOT [37] and NFS [38] to evaluate the performances of our tracker.

OTB100:

OTB100 is a popular benchmark for performance evaluation, including 100 short videos. The evaluation metrics are success rate and precision.

In this experiment, we compare our tracker with classical Siamese networks (e.g., SiamFC, SiamRPN++ and SiamBAN), online training methods (e.g., ATOM, DiMP, and MDNet) and some offline training methods. As shown in Table 2, our tracker ranks second in both success rate and precision.

Table 2. Comparative results on OTB100.

Trackers	SUCC.	PREC.
GradNet [3]	0.639	0.861
UpdateNet [23]	0.647	0.861
ATOM [22]	0.671	0.882
MDNet [27]	0.678	0.909
DSDCF [39]	0.667	0.784
DiMP [2]	0.688	0.900
ECO [5]	0.691	0.910
DaSiamRPN [4]	0.658	0.880
SiamFC [6]	0.587	0.722
SiamFC++ [40]	0.682	0.896
SiamBAN [17]	0.696	0.910
SiamRPN++ [12]	0.696	0.915
SiamRCNN [41]	0.700	0.891
DROL-RPN [42]	0.715	0.934
Ours	0.707	0.922

The evaluation metrics Succ and Prec. characterize the areas under the curves in Figure 5 below. Compared with the classical Siamese network SiamFC, we improve the success rate and precision by 20.4% and 27.7%, respectively. Compared with the recently published DSiamRPN, we also improve the two indexes by 7.4% and 4.8%, respectively. Meanwhile, we also attain some performance gain over excellent tracker SiamRPN++. Additionally, we surpass SiamRCNN [41] in success and precision by 1% and 4%. We only lag behind DROL-RPN [42]. However, we obtain better results than DROL-RPN for fast motion, low resolution, and large variation.

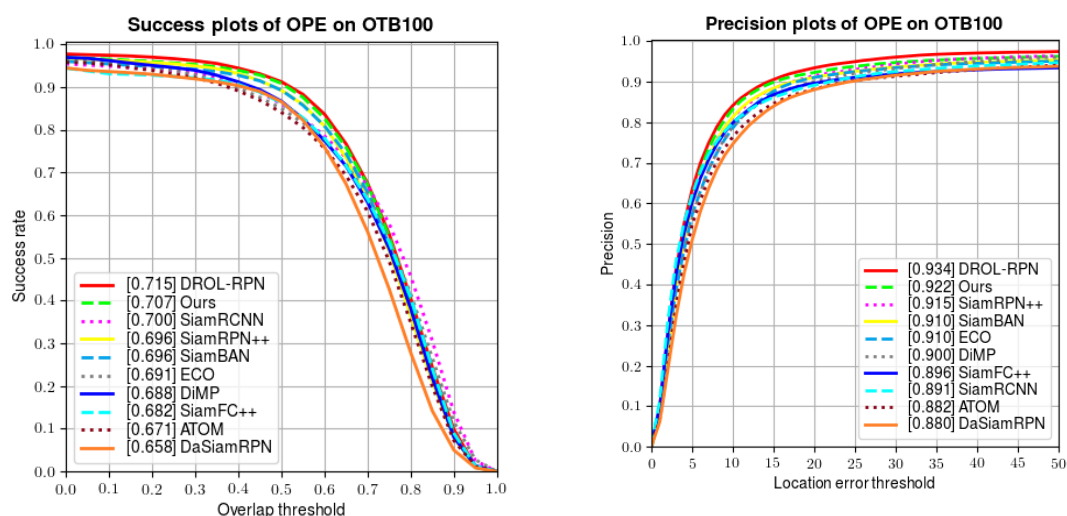


Figure 5. Comparative results of success rates and precision on OTB100.

The above experiment on OTB100 validates the effectiveness of our tracker, which proves the effect of an online training branch in improving the performance of original Siamese networks. Although classical Siamese networks extract useful target features, they are limited by the size of training sets and cannot adapt to target changes. Under such circumstances, our proposed online training branch remedies the above shortcomings by adjusting the weights during the training process to learn rich target features, so as to ensure the robustness of tracking.

On the other hand, compared with online-training-based trackers (including MDNet, ATOM, and DiMP), we also exceed them in success rates and precisions by some margins. The results demonstrate the fact that the Siamese network and online training branch are individually effective and are also complementary in boosting the tracking performance.

Apart from the above results, OTB100 also provides several attributes to test the robustness of each tracker in dealing with unpredictable challenges (including illumination variation, fast motion, scale variation, etc.). A series of comparisons are made in Figure 6 below; our tracker manifests favorable performances when handling fast motion, deformation, rotation, out-of-view, illumination variation, and scale variation, and also achieves a competitive performance in solving background clutter and low-resolution situations.

A qualitative comparison between different trackers on OTB100 is illustrated in Figure 7 below. The first video (the first row) involves target deformation, the second video (the second row) contains multiple similar objects, the third video (the third row) shows a rotating object, and the last video (the right row) is an example of small object. In these four situations, our tracker exhibits strong robustness against other popular methods when coping with adverse situations.

LaSOT:

Unlike OTB100, which has only short videos, LaSOT is a huge and widely used long-term dataset including 1400 videos, wherein each of them has more than 2500 frames on average. It also includes challenging situations such as background clutter, fast motion, scale variation, etc. We evaluate our tracker using 280 sequences on the test set of LaSOT. A quantitative comparison is made in Figure 8, showing the success rate, precision, and normalized precision of several trackers. Compared with both popular offline-training-based trackers (SiamFC [6], SiamBAN [17], SiamRPN++ [12], StrcutSiam [43], DSiam [44], UpdateNet [23], SiamRCNN [41], and Stark [33]) and online-training-based trackers (ATOM [22]), our tracker wins third place in all three metrics. The results on LaSOT demonstrate that our tracker excels at tracking tasks in long-term videos.

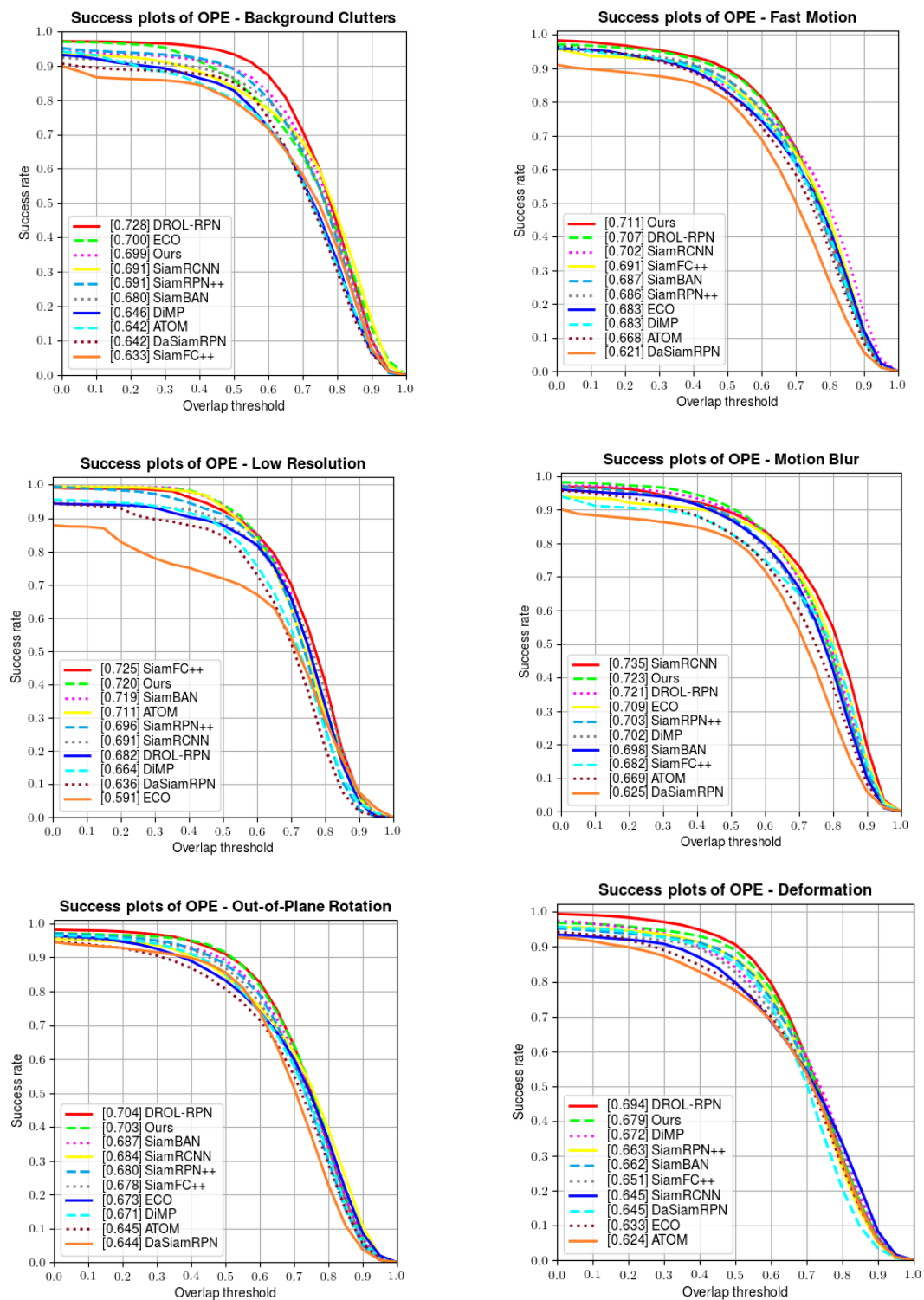


Figure 6. Cont.

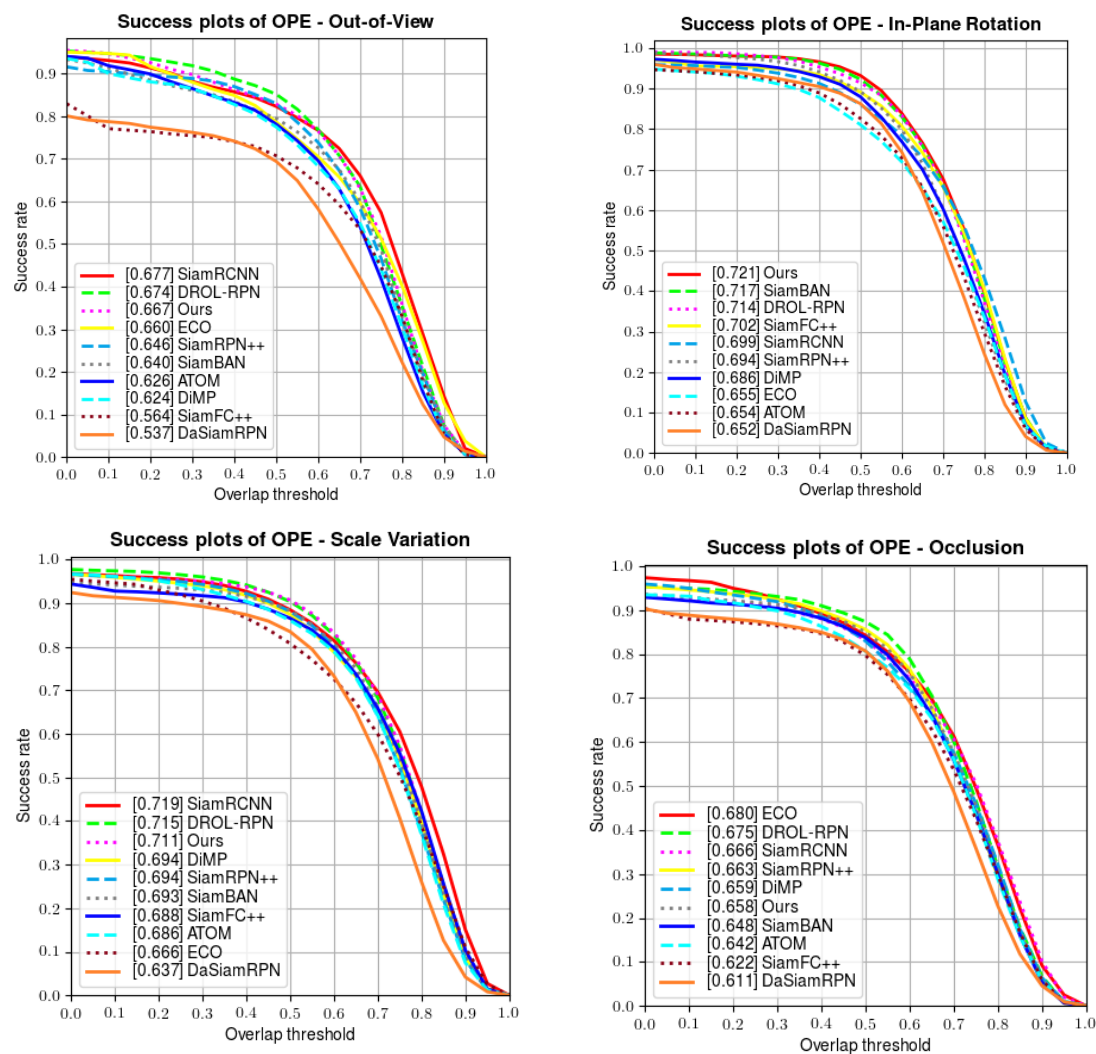
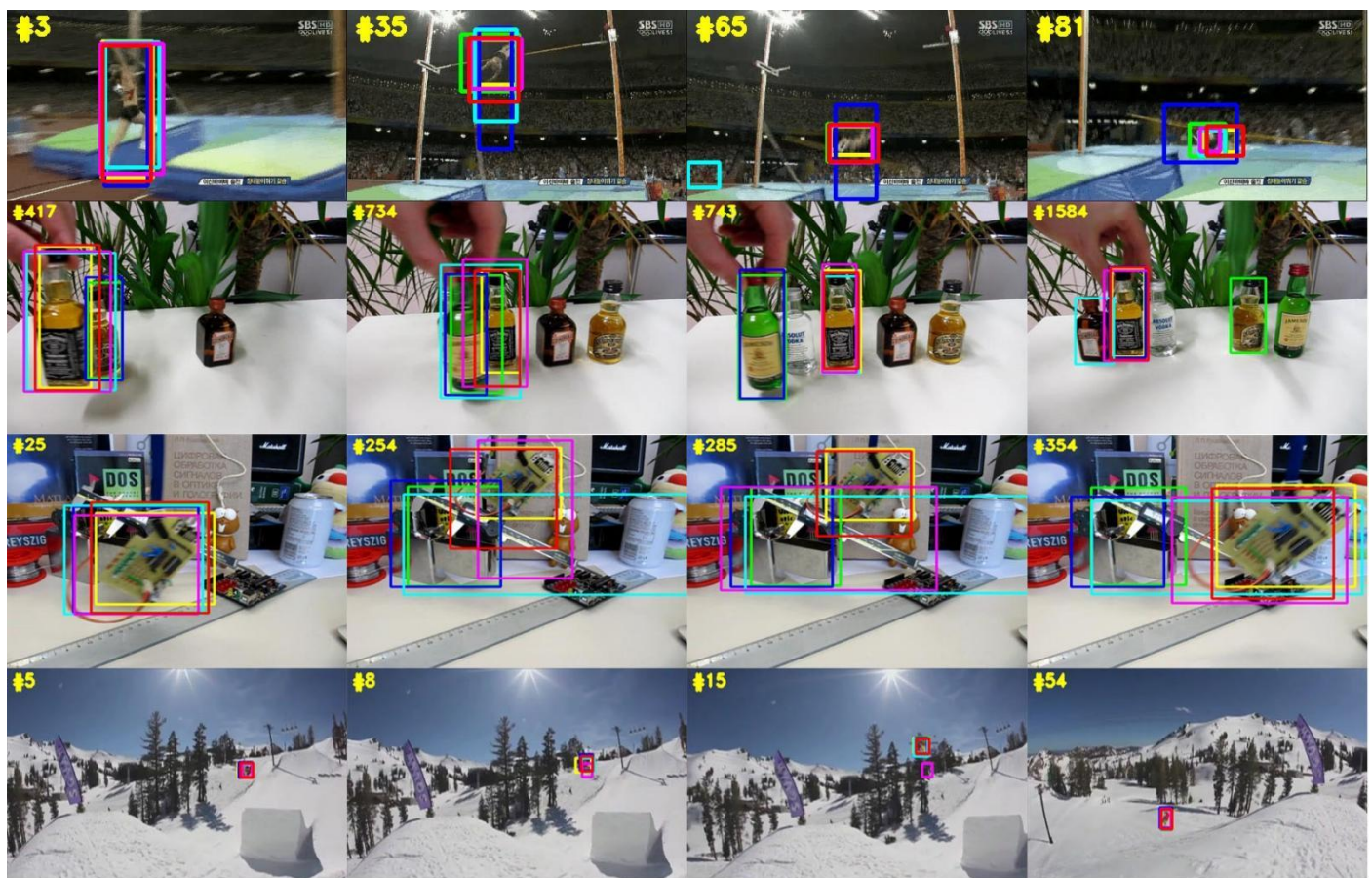


Figure 6. Comparisons of the attributes on OTB100.

Unlike the previous experiment on OTB100, in this experiment, we obviously lag behind Siam R-CNN and STARK. Our analysis of this result is as follows: the main innovation of Siam R-CNN is the combination of the re-detection module and the dynamic tracking algorithm. It measures the similarity between the region proposal and the template patch and updates the bounding box in a dynamic way, so that when the tracker loses the target, it regains the target in a short time. Similarly, STARK advised a score prediction head to update the template to ensure the robustness in the tracking process. In comparison, although our online module updates the template to capture target changes, it cannot re-detect the target when it disappears temporarily. We think an additional local search strategy would help with target re-detection.

NFS:

NFS contains 100 videos of real scenes captured by higher-speed cameras, in which we select NFS30 (videos with 30 fps) for testing. The evaluation metrics are similar to OTB100. The comparative results are listed in Table 3 above.



GT □ SiamOT □ SiamBAN □ SiamRPN++ □ ATOM □ DiMP □

Figure 7. Qualitative comparison of our tracker with other popular trackers on 4 video sequences from OTB100.

Table 3. Comparative results on NFS.

Trackers	SUCC.
UPDT [35]	0.537
MDNet [27]	0.422
C-COT [45]	0.488
ECO [5]	0.466
ATOM [22]	0.584
DiMP [2]	0.620
SiamBAN [17]	0.594
Ours	0.630

As expected, our tracker again ranks first in both success rate and precision on NFS among all the popular methods. Since the videos in NFS are all from real life (YouTube life videos), the results on NFS have high practical significance.

5.1. Ablation Study

In Section 3.2.1, we discussed multiple feature fusion schemes, including addition, concatenation, multiplication, and weighted fusion. In this section, we test and compare different fusion schemes and attention mechanisms along with key parameters to analyze the optimal configuration.

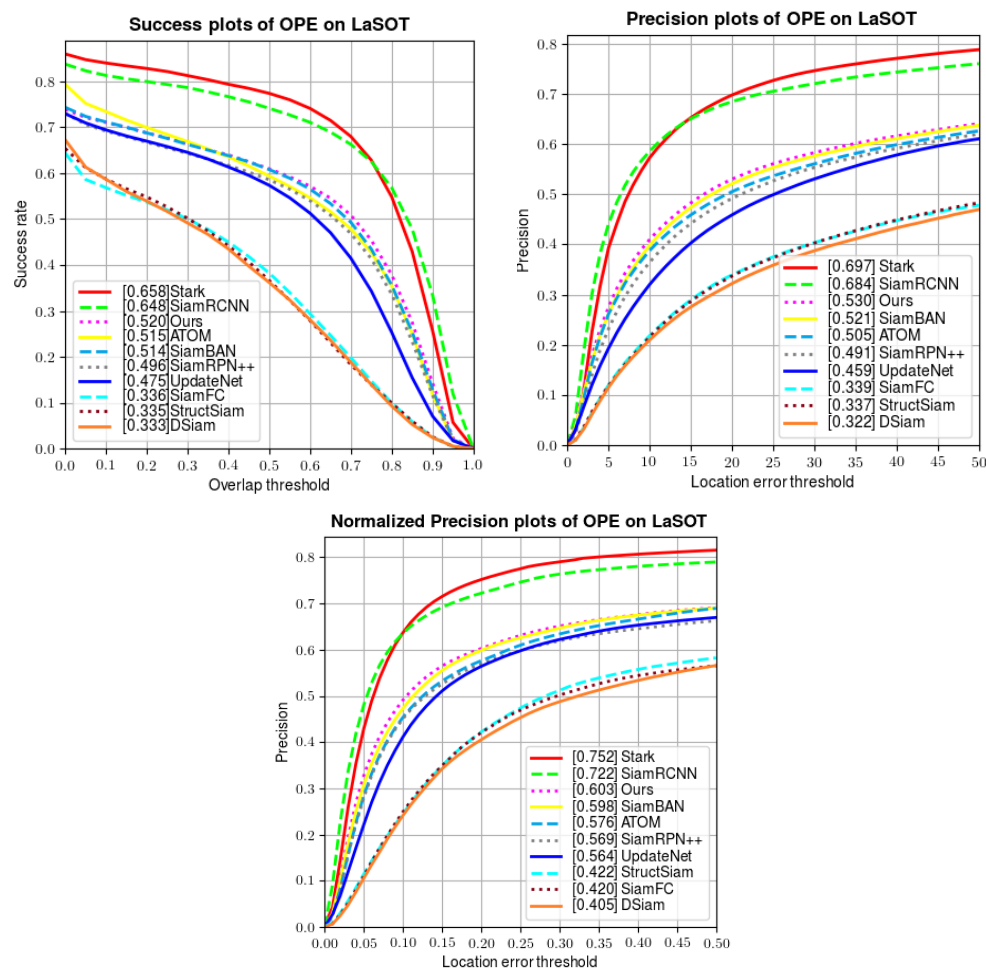


Figure 8. Comparison of tracking performance on LaSOT.

5.1.1. Fusion Scheme

We investigate the following feature fusion schemes (expressed as Equations (3)–(6)) to extract rich semantic features, and search for the best strategy:

As shown in Table 4, the weighted fusion method (Equation (6)) has the highest success rate and precision. We thereby utilize weighted fusion to integrate the two branches.

Table 4. Comparison of different feature fusion methods.

Fusion Strategy	SUCC.	PREC.
Addition	0.704	0.916
Concatenation	0.706	0.915
Multiplication	0.704	0.920
Weighted Fusion	0.707	0.922

5.1.2. Attention Mechanism

In this section, we test different combinations of attention mechanisms on a NFS30 dataset to testify their effectiveness. As shown in Table 5, without any attention modules, the success rate is only 0.503. When different attention modules are added into the network, the success rate increases significantly. Finally, when we add self-attention, spatial attention, and channel attention modules, we achieve the highest success rate.

Table 5. Comparison of results of different attention mechanisms on NFS.

Attention Mechanism	SUCC.
None	0.503
Self-Attention	0.565
Self-Attention + Spatial Attention	0.589
Self-Attention + Channel Attention	0.591
Self-Attention + Spatial Attention + Channel Attention	0.630

5.1.3. The Weights in Fusion Process

In this section, we test different values of parameter η , which balances the weights between the Siamese network and the online training branch. We believe the target state in the online branch should be given higher weights than the classification branch (i.e., $\eta > 0.5$). To search for the optimal value of η , we list different success and precision results using different values of η (shown in Table 6). Fortunately, when $\eta = 0.8$, we obtain the highest success and precision values, which confirms our conjecture. Therefore, we select $\eta = 0.8$ in the weighted fusion process.

Table 6. The comparative results of success and precision using different weights.

η	SUCC.	PREC.
0.1	0.693	0.907
0.2	0.688	0.896
0.3	0.698	0.909
0.4	0.704	0.919
0.5	0.705	0.911
0.6	0.706	0.921
0.7	0.701	0.912
0.8	0.707	0.922
0.9	0.701	0.915

6. Conclusions

In this paper, we develop a Siamese-based tracking network integrated with an online training branch, where the core components include feature fusion and improved attention modules. Our proposed architecture not only extracts rich target features to ensure tracking accuracy and efficiency, but also proves that the classical Siamese network and the online training strategy could be organically combined and are complementary in tracking tasks. Extensive experiments have been conducted on three prevailing tracking benchmarks. The experimental results verify the effectiveness and advancement of our tracker compared with other popular methods. We hope our work will provide a reference for those who are studying Siamese networks for visual tracking. Meanwhile, we are now investigating the combination of convolution blocks and transformer structure, and plan to apply it to visual tracking in the near future.

Author Contributions: Formal analysis, Y.Z. (Yi Zhang); Investigation, Y.Z. (Yi Zhang); Methodology, Y.Z. (Yuxin Zhou); Project administration, Y.Z. (Yi Zhang); Validation, X.G.; Visualization, X.G.; Writing—original draft, Y.Z. (Yi Zhang). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13–18 June 2010.
2. Bhat, G.; Danelljan, M.; Van Gool, L.; Timofte, R. Learning discriminative model prediction for tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, Seoul, Korea, 27 October–2 November 2019; pp. 6181–6190.
3. Li, P.; Chen, B.; Ouyang, W. GradNet: Gradient-guided network for visual object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6162–6171.
4. Zhu, Z.; Wang, Q.; Li, B. Distractor-aware Siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 101–117.
5. Danelljan, M.; Bhat, G.; Shahbaz Khan, F. ECO: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
6. Bertinetto, L.; Valmadre, J.; Henriques, J.F. Fully-convolutional Siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 850–865.
7. Wang, G.; Luo, C.; Xiong, Z. SPM-Tracker: Series-parallel matching for real-time visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3643–3652.
8. Li, B.; Yan, J.; Wu, W. High performance visual tracking with Siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
9. Ren, S.; He, K.; Girshick, R. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Processing Syst.* **2015**, *28*. [[CrossRef](#)] [[PubMed](#)]
10. Wang, Q.; Zhang, L.; Bertinetto, L. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1328–1338.
11. Fan, H.; Ling, H. Siamese cascaded region proposal networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7952–7961.
12. Li, B.; Wu, W.; Wang, Q. SiamRPN++: Evolution of Siamese visual tracking with very deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
13. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4591–4600.
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
15. Tang, C.; Qin, P.; Zhang, J. Robust template adjustment Siamese network for object visual tracking. *Sensors* **2021**, *21*, 1466. [[CrossRef](#)] [[PubMed](#)]
16. Zeng, Y.; Zeng, B.; Yin, X.; Yang, W. SiamPCF: Siamese point regression with coarse-fine classification network for visual tracking. *Appl. Intell.* **2021**, *52*, 4973–4986. [[CrossRef](#)]
17. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese box adaptive network for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6668–6677.
18. He, A.; Wang, G.; Luo, C. AF2S: An Anchor-Free Two-Stage Tracker Based on a Strong SiamFC Baseline. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 637–652.
19. Zhang, Z.; Peng, H.; Fu, J. Ocean: Object-aware anchor-free tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 771–787.
20. Peng, S.; Yu, Y.; Wang, K. Accurate anchor free tracking. *arXiv* **2020**, arXiv:2006.07560.
21. Held, D.; Thrun, S.; Savarese, S. Learning to track at 100 fps with deep regression networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 749–765.
22. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate tracking by overlap maximization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4660–4669.
23. Zhang, L.; Gonzalez-Garcia, A.; van de Weijer, J.; Danelljan, M.; Khan, F.S. Learning the Model Update for Siamese Trackers. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 4009–4018. [[CrossRef](#)]
24. Yang, T.; Chan, A.B. Learning dynamic memory networks for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 152–167.
25. Yang, T.; Chan, A.B. Recurrent filter learning for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 2010–2019.
26. Yang, T.; Chan, A.B. Visual tracking via dynamic memory networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 360–374. [[CrossRef](#)]
27. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
28. Zhang, Y.; Wang, D.; Wang, L. Learning regression and verification networks for long-term visual tracking. *arXiv* **2018**, arXiv:1809.04320.

29. Yao, Y.; Wu, X.; Zhang, L. Joint representation and truncated inference learning for correlation filter based tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 552–567.
30. Choi, J.; Kwon, J.; Lee, K.M. Real-time visual tracking by deep reinforced decision making. *Comput. Vis. Image Underst.* **2018**, *171*, 10–19. [[CrossRef](#)]
31. Park, E.; Berg, A.C. Meta-tracker: Fast and robust online adaptation for visual object trackers. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 569–585.
32. Hu, X.; Liu, H.; Hui, Y. Transformer feature enhancement network with template update for object tracking. *Sensors* **2022**, *22*, 5219. [[CrossRef](#)] [[PubMed](#)]
33. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning Spatio-temporal transformer for visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10448–10457.
34. Wang, N.; Zhou, W.; Wang, J. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1571–1580.
35. Bhat, G.; Johnande Danelljan, M. Unveiling the power of deep tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 483–498.
36. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
37. Fan, H.; Lin, L.; Yang, F. Lasot: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5374–5383.
38. Galoogahi, H.K.; Fagg, A.; Huang, C.; Ramanan, D.; Lucey, S. *Need for Speed: A Benchmark for Higher Frame Rate Object Tracking*; ICCV: Venice, Italy, 2017; pp. 1134–1143.
39. Gao, L.; Liu, B.; Fu, P. Visual tracking via dynamic saliency discriminative correlation filter. *Appl. Intell.* **2022**, *52*, 5897–5911. [[CrossRef](#)]
40. Xu, Y.; Wang, Z.; Li, Z. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12549–12556. [[CrossRef](#)]
41. Voigtlaender, P.; Luiten, J.; Torr, P.H.; Leibe, B. Siam R-CNN: Visual tracking by re-detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6578–6588.
42. Zhou, J.; Wang, P.; Sun, H. Discriminative and robust online learning for Siamese visual tracking. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 13017–13024. [[CrossRef](#)]
43. Zhang, Y.; Wang, L.; Qi, J. Structured Siamese network for real-time visual tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 351–366.
44. Guo, Q.; Feng, W.; Zhou, C. Learning dynamic Siamese network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1763–1771.
45. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 472–488.