Engineering Applications of Artificial Intelligence

A Spatial-Temporal Contexts Network for Object Tracking

--Manuscript Draft--

Manuscript Number:	EAAI-23-3779
Article Type:	Research paper
Keywords:	Object Tracking; Spatial and Temporal Contexts; ConvLSTM; Similarity Map Fusion
Corresponding Author:	Jun Chu, Prof. Nanchang Hangkong University CHINA
First Author:	Kai Huang
Order of Authors:	Kai Huang
	Kai Xiao
	Jun Chu
	Lu Leng
	Xingbo Dong
Abstract:	In recent years, although there have been significant advancements and developments in visual object tracking, most trackers still fail to adapt to the deterioration of object appearance in complex scenes. They typically only utilize spatial information or use simple temporal networks. The fusion of spatial and temporal contexts among consecutive frames can hypothetically capture historical information to boost the tracking performance but inevitably pollutes the model with noisy samples. To that end, we propose a novel end-to-end ConvLSTM-based tracking framework that uses spatial and temporal information from each frame and adapts to noisy samples, called STCTracker. Specifically, a multilayer residual ConvLSTM-based Spatial-Temporal Contexts Network (STCN) is proposed in STCTracker to retain the target's past information to guide the tracker to focus on the most informative regions of the current frame. Furthermore, a multi-similarity map fusion model is proposed to calculate the pixel-level similarities map, allowing STCTracker to retrieve historical target information from different time adaptively and be resilient to partial occlusions and non-rigid deformations. Extensive empirical studies are carried out on benchmarks including OTB2015, GOT-10K, TrackingNet, LaSOT, UAV123, and VOT2018. The empirical results suggest that our STCTracker achieves state-of-the-art performance compared with existing schemes. We make our code and models publicly available to encourage further research on this topic: www.github.com/Kevoen/STCTrack.
Suggested Reviewers:	Bing Li bing.li@ia.ac.cn
	Mantun Gao gaomant@nwpu.edu.cn

Declaration of interests

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

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jun Chu reports financial support was provided by National Natural Science Foundation of China (No.62162045). Lu Leng reports financial support was provided by Technology Innovation Guidance Program Project (Special Project of Technology Cooperation, Science and Technology Department of Jiangxi Province, No.20212BDH81003). Lu Leng reports financial support was provided by National Natural Science Foundation of China (No.61866028).

Highlights

A Spatial-Temporal Contexts Network for Object Tracking

Kai Huang, Kai Xiao, Jun Chu, Lu Leng, Xingbo Dong

- A ConvLSTM-based Spatial-Temporal Contexts Network (STCN) is proposed to trace the prior knowledge of the target's spatial and temporal contexts explicitly by fusing information from previous frames with the current frame's feature map. Furthermore, by residual connecting the current feature map with the prior knowledge, the network is better equipped to handle appearance changes caused by non-rigid deformations and occlusions.
- A multi-similarity map fusion module (Multi-SMFM) is designed to construct pixel-level similarity maps for target localization by jointly utilizing historical target information from different times while eliminating the noise in the video.
- On top of the STCN and Multi-SMFM, a novel end-to-end object tracker, namely STCTracker, is established. STCTrack integrates spatial and temporal contexts in an efficient and effective way by utilizing a ConvLSTM-based network (STCN) and a multi-similarity map fusion module (Multi-SMFM). Extensive experiments on public benchmark datasets demonstrate the effectiveness and efficiency of STCTrack compared to existing state-of-the-art tracking algorithms.

A Spatial-Temporal Contexts Network for Object Tracking

Kai Huang[a], Kai Xiao[a], Jun Chu[a], Lu Leng[a], Xingbo Dong[b]

[a]*Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang, 330063, China*
[b]*School of Artificial Intelligence, Anhui University, Anhui, 230093, China*

Abstract

In recent years, although there have been significant advancements and developments in visual object tracking, most trackers still fail to adapt to the deterioration of object appearance in complex scenes. They typically only utilize spatial information or use simple temporal networks. The fusion of spatial and temporal contexts among consecutive frames can hypothetically capture historical information to boost the tracking performance but inevitably pollutes the model with noisy samples. To that end, we propose a novel end-to-end ConvLSTM-based tracking framework that uses spatial and temporal information from each frame and adapts to noisy samples, called STCTracker. Specifically, a multilayer residual ConvLSTM-based Spatial-Temporal Contexts Network (STCN) is proposed in STCTracker to retain the target's past information to guide the tracker to focus on the most informative regions of the current frame. Furthermore, a multi-similarity map fusion model is proposed to calculate the pixel-level similarities map, allowing STCTracker to retrieve historical target information from different time adaptively and be resilient to partial occlusions and non-rigid deformations. Extensive empirical studies are carried out on benchmarks including OTB2015, GOT-10K, TrackingNet, LaSOT, UAV123, and VOT2018. The empirical results suggest that our STCTracker achieves state-of-the-art performance compared with existing schemes. We make our code and models publicly available to encourage further research on this topic: www.github.com/Kevoen/STCTrack.

Keywords: Object Tracking, Spatial and Temporal Contexts, ConvLSTM, Similarity Map Fusion

1. Introduction

Object tracking is a major computer vision research topic attracting significant attention over the decades. It has been applied to various applications such as human-computer interactions [1], video surveillance [2, 3], and autonomous driving [4, 5, 6]. In object tracking, the appearance of the target usually deteriorates due to various factors such as motion blur, background clutter, non-rigid deformation, or partial occlusion, which poses a significant challenge to the target tracker. Utilizing historical spatial and temporal features jointly could theoretically improve tracking performance as that information provides essential clues about the objects' scenes, appearances, and motion across the video frames. Trackers often use online updates, template updates, and the introduction of spatio-temporal information to improve tracking performance.

Siamese network based trackers [7, 8, 9, 10] usually do not perform template updates or use simple linear calculations for template updates. However, template update strategy can accumulate errors over time, making it difficult to adapt to template appearance changes caused by occlusion, non-rigid deformation, and other factors. To address this issue, UpdateNet [11] updates the template by learning a template tracking network. Although it solves the singularity of template updates and improves performance, the frame-by-frame updates involve a lot of redundant calculations. Most offline Siamese network-based trackers rely only on spatial information and ignore the historical information on the temporal dimension.

However, some correlation filters based trackers [12, 13] use online updating mechanisms to achieve stronger robustness than Siamese network-based trackers. These methods require gradient calculations during the inference stage to update layers in a specific domain using gradient descent. These methods use gradient update strategies including Gauss-Newton/meta-learning, but many devices do not support backpropagation, greatly limiting the application of gradient-based methods. At the same time, online template updating requires more computational resources, which poses a challenge for real-time tracking.

Furthermore, some methods [14, 15] use memory networks to leverage spatial and temporal information to improve tracking performance. MemTrack [14] is one of the typical representatives, which uses a memory network to read a residual template during the tracking process, and then combines it with the initial template to generate a composite template as the up-

dated representation of the target. Although MemTrack uses a lot of historical information, the memory read operation controlled by LSTM may lose useful information. Some methods use single-layer ConvLSTM to characterize the historical information, but its simple network structure cannot adapt to the complex appearance changes of the target. STMTrack [15] proposes a spatiotemporal memory network-based tracking framework that can fully leverage the historical information relevant to the target, thereby better adapting to the target appearance changes during the tracking process. Although STMTrack [15] introduces a memory mechanism to store the target's historical information, the target's historical information contains a lot of redundant information, and the mechanism only stores the historical appearance information of the target, ignoring the motion information.

We coherently deal with the aforementioned issues by proposing the STC-Tracker, an end-to-end tracking model consisting of a standard feature extractor network, a novel ConvLSTM-based Spatial-Temporal Contexts Network (STCN), a new multi-similarity map fusion module (Multi-SMFM) and a head network.

Different from [14, 15], our proposed Spatial-Temporal Context Network (STCN) is a multilayer residuals network based on ConvLSTM. Compared with single-layer ConvLSTM, multilayer residuals STCN has stronger ability to model the appearance of the target. Unlike STMTrack which directly utilizes the history information of each frame, the STCN module reduces redundant computations and improves robustness by using the history information to model the appearance of the target.

Object appearance across frames may suffer from noisy intra-class variations; hence historical information from prior frames must be handled adaptively based on the information in the current frame. Besides, directly concatenating previous feature maps generated from STCN will lead to a suboptimal outcome due to the noise in the video. Therefore, we propose a multi-similarity map fusion module (Multi-SMFM) to construct the pixel-level similarities map for target localization. Multi-SMFM allows STCTrack to obtain historical target information from different times jointly while eliminating the influence of noise. Using the Multi-SMFM, an interaction between the spatial and temporal contexts can be established and can be utilized to improve the similarity map's robustness to variations.

The main contributions of this work are summarized as follows:

1. A ConvLSTM-based Spatial-Temporal Contexts Network (STCN) is

proposed to trace the prior knowledge of the target's spatial and temporal contexts explicitly by fusing information from previous frames with the current frame's feature map. Furthermore, by residual connecting the current feature map with the prior knowledge, the network is better equipped to handle appearance changes caused by non-rigid deformations and occlusions.

2. A multi-similarity map fusion module (Multi-SMFM) is designed to construct pixel-level similarity maps for target localization by jointly utilizing historical target information from different times while eliminating the noise in the video.
3. On top of the STCN and Multi-SMFM, a novel end-to-end object tracker, namely STCTracker, is established. STCTrack integrates spatial and temporal contexts in an efficient and effective way by utilizing a ConvLSTM-based network (STCN) and a multi-similarity map fusion module (Multi-SMFM). Extensive experiments on public benchmark datasets demonstrate the effectiveness and efficiency of STCTrack compared to existing state-of-the-art tracking algorithms.

2. Related works

2.1. Visual Object Tracking

In tracking-by-detection methods, the target is distinguished from the background using an online learning classifier, and object tracking is viewed as a detection task inside a region-of-interest (ROI) image. MDNet [16] uses large amounts of annotated tracking data to learn an online classifier for each object using a multi-domain learning strategy. SANet [17] proposes a structure-aware network to manage similar distractions. ATOM [13] improves its tracking of targets using discriminatory correlation filters and IoU-Net. Even though these methods have achieved remarkable tracking performance, they suffer from poor efficiency as they need to extract features from a large number of bounding box proposals and online fine-tune.

Using a pre-trained network, matching-based tracking methods [7, 8, 9, 10] were proposed to match candidate proposals with target templates. Typically, these approaches do not involve any online update operations; therefore, they offer considerable speed benefits. Recently, siamese-based trackers have distinguished themselves for their outstanding accuracy and attractive effectiveness [18, 7, 8, 9, 10]. The majority of offline Siamese

trackers [18, 19, 20] are solely spatial trackers that consider object tracking as a template-matching between the original template and the current search region. Most trackers apply variants of correlation, including naive correlation [21, 18], depth-wise [9, 22] correlation, and point-wise [20, 23] correlation, to extract the relationship between templates and search locations along the spatial dimension.

Despite achieving state-of-the-art performance, historical spatial and temporal information is not fully-utilized for tracking, which hampers the performance gain. Differently, our tracker can effectively model the historical spatial and temporal contexts during tracking to increase the discriminability and robustness.

2.2. Spatial-Temporal Models for Tracking

Both spatial and temporal information is essential for object tracking. Spatial information describes the object's appearance, while temporal information describes the object's state change between frames. CLRST [24] uses temporal consistency to prune and choose candidate particles adaptively. By regularly updating the CNNs that are stored in the CNN pool, DeepTrack [25] demonstrates temporal adaptability. RTT [26] uses spatial-temporal information in the form of long-range contextual cues. By training a recursive tracking network, Re3 [27] adds temporal information to the model and transforms the image embedding into a bounding box. LTMU [28] trains a meta-updater to predict whether the current state is reliable enough to be utilized for long-term tracking updates.

Recent studies [14, 29] try to employ temporal information for better object feature representation. RFL[30] uses a recurrent neural network (RNN) to estimate an object-specific filter for tracking. However, it uses only a simple ConvLSTM as a filter, and its ability to model the appearance of the target is weak. In order to update the feature representation model, MemTrack [14] proposes a dynamic memory network, the memory reading operation under LSTM control may result in the loss of useful information. RecTrack[31] employs a gated recurrent unit to represent the temporal information of sequences, whereas RATM[29] incorporates an attention mechanism into the RNN to assist in the search for a target.

STMTrack [15] proposes a new tracking framework that uses a space-time memory network to adapt to appearance changes during tracking. The framework introduces a novel memory mechanism to store historical information of the target and guide the tracker to focus on informative regions in

the current frame. Additionally, the pixel-level similarity computation of the memory network improves the accuracy of the bounding box of the target. The proposed tracker outperforms previous state-of-the-art real-time methods on multiple challenging benchmarks, such as OTB-2015, TrackingNet, GOT-10k, LaSOT, UAV123, and VOT2018, while running at 37 FPS.

Despite the impressive performance achieved by the aforementioned works, there are still some issues that need to be addressed. For instance, the use of RNNs, due to their simplistic network architecture, has limited ability to effectively model the appearance, location, and temporal movement of targets, resulting in poor accuracy in bounding box predictions. While STMTrack [15] attempts to address this issue by using a space-time memory network to adapt to appearance changes during tracking, its direct utilization of historical information from each frame suffers from redundancy in information computation.

Unlike existing methods, our approach addresses the issue of poor modeling capability by utilizing a residual network consisting of multiple layers of ConvLSTMs, referred to as the spatio-temporal contexts network, which is capable of accurately modeling the appearance of the target. While our work and STMTrack [15] share a common concept of spatial and temporal modeling, we differentiate ourselves by utilizing ConvLSTM to explicitly capture the interdependence between spatial and temporal information across previous frames. To improve the accuracy of target bounding box prediction, we employ a multi-similarity map fusion module. This framework naturally accommodates the target's evolutionary process in a video series by effectively utilizing historical data about the target, thereby accommodating changes in the target's appearance during tracking.

3. Methodology

3.1. Overview

As illustrated in Fig. 1, the framework is composed of four modules: a feature extraction network, a spatial-temporal contexts network, a multi-similarity map fusion module, and a head network. The feature extraction network consists of a historical template branch and a search branch. Both historical frames and corresponding foreground-background label (fb_label) maps are inputs for the historical template branch. In this work, the search branch is the current frame in a video sequence, and the historical frames

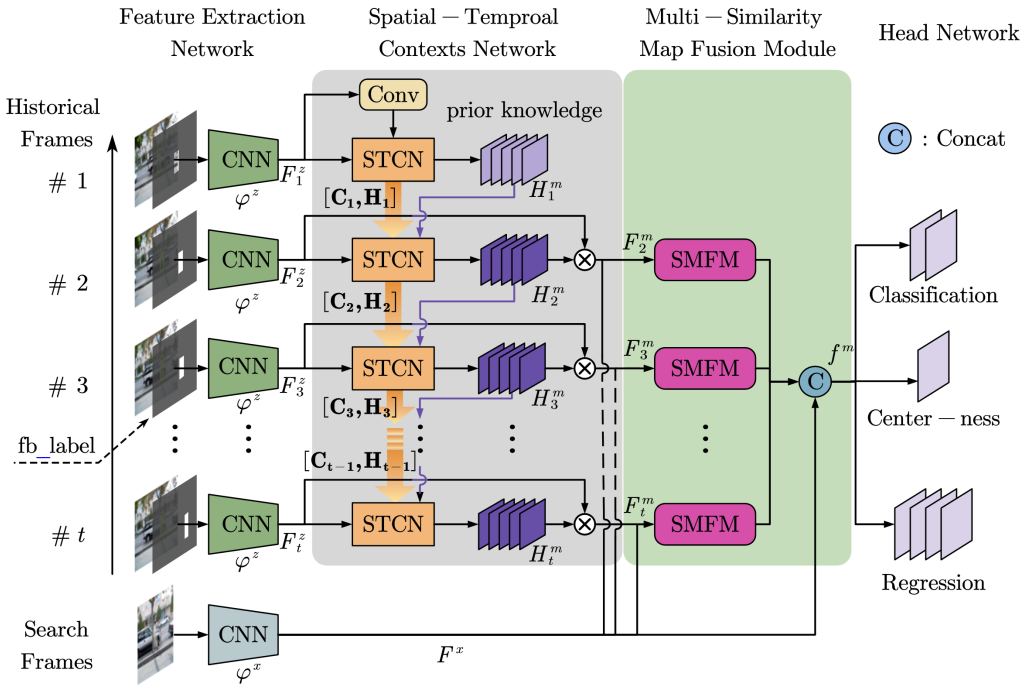


Figure 1: Overview of our architecture. The architecture of our system consists of four main components: 1) the feature extraction network located on the left side of the diagram, and it has two branches: a search branch and a history template branch; 2) Spatial-Temporal Contexts Network (STCN) for online spatial and temporal feature extraction shown in Fig. 2 (b); 3) Multi-Similarity Map Fusion Module for similarity historical features fusion shown as Fig. 4, and 4) the head networks for classification and target bounding box regression. The inputs for the History Template branch are the history frames and the associated foreground-background label mappings (fb_label).

are multiple historical template frames. After feature extraction, the spatial-temporal contexts network captures the feature relationships between each frame in the sequence. It reinforces the original features with historical spatial-temporal contextual information. Then, the similarity map fusion model extracts similarity information pertaining to the target from the reinforced feature maps of all previous frames and builds a similarity fusion map to distinguish the target from the background and predict the target bounding box for the search frame.

3.2. Feature Extraction Network

A feature extraction network (GoogleNet is adopted in this work) is adopted for both the history frames branch and the search branch. For the history frames branch, t historical frames are fed to the feature extraction network. However, directly using extracted features from each frame will introduce noisy information to the subsequent temporal modeling process of the target object. Therefore, inspired from [15], we introduce t foreground-background label map to avoid noisy interference.

Specifically, we mark each pixel as 1 in the corresponding ground truth bounding box, and 0 at other locations, as a foreground-background label m_i for each history frame h_i, where $i \in [0, t]$. We utilize two convolutional layers η, θ, to embed the history frame h_i and label map m_i into the same space, respectively. Then, we element-wise-add $\eta(h_i)$ and $\theta(m_i)$. After that, we input the sum to backbone φ^z to extract the t history feature map F, which denoted each history feature map F_i^t and computed as:

$$F_i^t = \varphi^z(\eta(h_i) \oplus \theta(m_i)). \tag{1}$$

where $F_i^t \in \mathbb{R}^{C \times H \times W}$ and $\oplus$ is element-wise-addition.

Compared to the history frames branch, the search branch takes only one search frame x as the input and generates a feature map F^x:

$$F^x = \varphi^x(x), \tag{2}$$

where $F^x \in \mathbb{R}^{C \times H \times W}$, φ^x represents all layers of the backbone.

3.3. Spatial-Temporal Contexts Network

ConvLSTM networks have achieved remarkable results in applications such as video prediction [32] and video comprehension [33]. It can learn to capture content and dynamic spatial-temporal representations. Inspired by this observation, we construct STCN that explicitly exploits the long-term Spatial-temporal information of the target object.

Fig. 2 depicts the design of the ConvLSTM cells utilized by the proposed Spatial-Temporal Contexts Network (STCN). The ConvLSTM can be defined as follows:

$$i_t = \sigma\left(W_{xi} * F_t^i + W_{hi} * H_{t-1}^i + b_i\right), \tag{3}$$

$$f_t = \sigma\left(W_{xf} * F_t^i + W_{hf} * H_{t-1}^i + b_f\right), \tag{4}$$

$$o_t = \sigma\left(W_{xo} * F_t^i + W_{ho} * H_{t-1}^i + b_o\right), \tag{5}$$

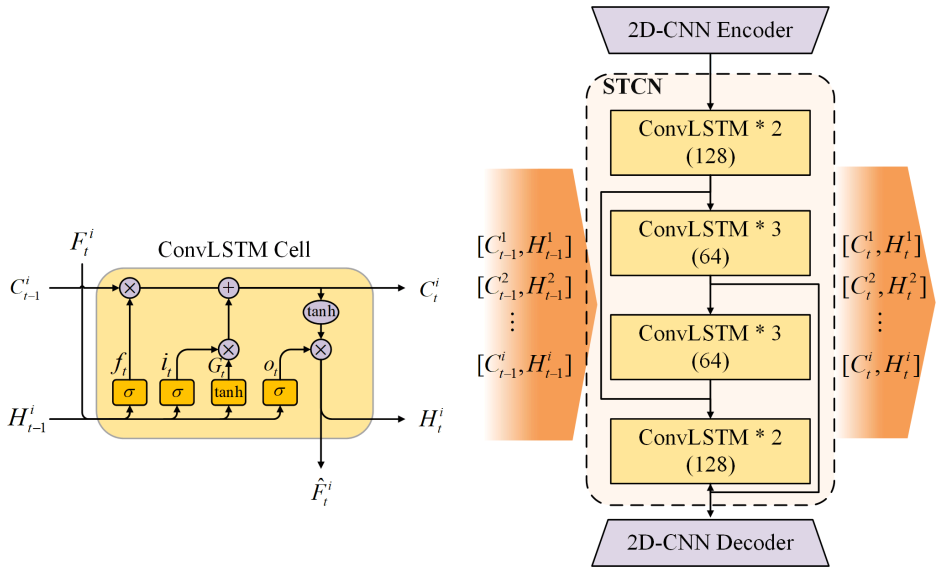


Figure 2: The architecture of a Spatial-temporal Contexts Network. (a) a ConvLSTM Cell (left). Besides a hidden cell H and a state cell C, a ConvLSTM cell contains an input gatei_t, a forget gate f_t, a candidate memory G_t, and an output gate o_t. (b) the architecture of the proposed Spatial-Temporal Contexts Network(right). The 2D-CNN encoder restores the resolution of feature maps, $[C, H]$ is the historical spatial memory cell and hidden state cell, which process the spatial representations of a template.

$$G_t = \tanh\left(W_{xc} * F_t^i + W_{hc} * H_{t-1}^i + b_c\right), \tag{6}$$

where σ is the sigmoid function, $W_{x\sim}$ and $W_{h\sim}$ are 2D convolution kernels and the weights of forgetting gate, input gate, and output gate, respectively. b_f, b_i, b_c and b_o are the biases for each gate. The input is F_t^i and the hidden state is H_t^i. The candidate memory G_t, and the gate i_t, f_t, o_t are all 3D tensors. $*$ is the convolution operator. The i represents the i-th layer of the STCN, and t is the number of frames in a sequence.

The memory cell C_t^i is trained to selectively forget and memorize the past information and the current input information, while the output gate o_t is trained how much of the memory cell to transmit to the hidden cell H_t^i. Given inputs F_t^i, H_{t1}^i and C_{t1}^i, the ConvLSTM cell updates the hidden cell and memory cell to H_t^i and C_t^i as:

$$C_t^i = F_t^i \circ C_{t-1}^i + i_t \circ G_t, \tag{7}$$

$$H_t^i = o_t \circ \tanh\left(C_t^i\right). \tag{8}$$

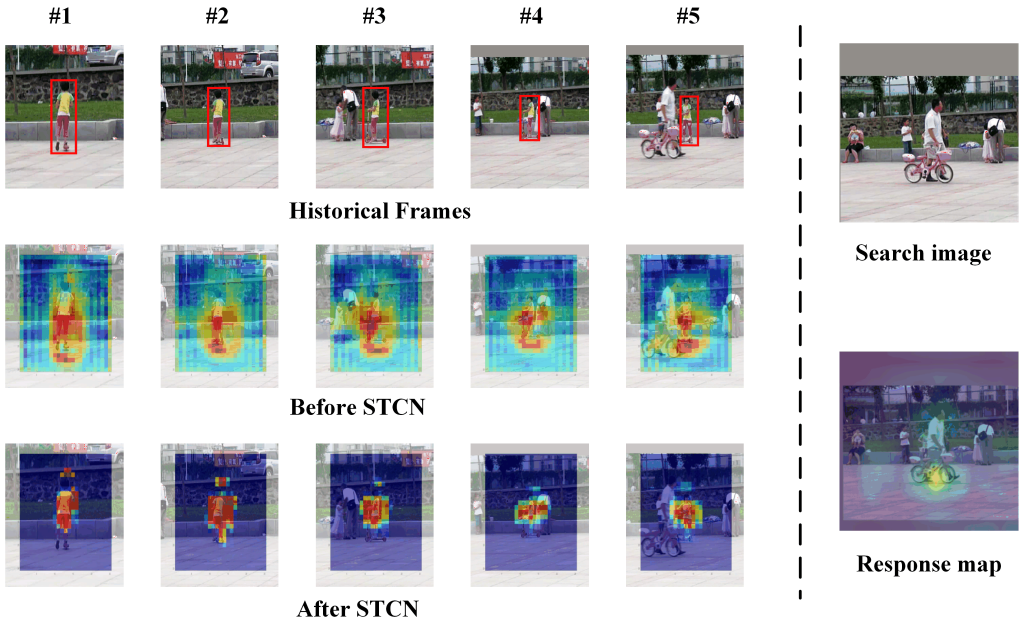


Figure 3: A comparison of the feature maps extracted before STCN (the second row) and after STCN (the third row).

where $\circ$ is the Hadamard product.

The architecture of the proposed STCN is given in Fig. 2 (b). The network architecture consists of a 2D-encoder-decoder and a stack of ConvLSTM. Both 2D-encoder and 2D-decoder contain one 2d convolutional layer and a normalized layer, which encodes the image from the resolution of 512×512 to 128×128, and restores the resolution from 128×128 to 512×512, respectively. Then, the proposed STCN uses a stack of 10 layers of ConvLSTM. The first and last layers have 128 channels, and the 6 middle layers have 256 channels. According to [32], two skip connections are established between the layers (2, 8) and (5, 10) based on channel concatenation.

As illustrated in Fig. 1, the STCN extracts spatial and temporal prior knowledge by combining past information with the current feature. In order to prevent the loss of original information, the final spatial and temporal feature for the current step is obtained by residual connecting with spatial-temporal prior knowledge and the current extracted feature map. The structure of the STCN is presented in Fig. 1 and Fig. 2(b).

Because of the large inter-class variations, employing a uniform initialization for the first spatial and temporal prior H_0^m would be inappropriate. We establish the initial spatial and temporal priors via a convolution layer over the initial feature map F_0, i.e., $H_0^m = \mathcal{F}_{init}(F_1)$. Given the current feature map F_t and the previous spatial and temporal knowledge H_{t-1}^m, the final output F_t^m can be obtained as:

$$F_t^m = F_t \otimes STCN(Concat(F_t, H_{t-1}^m)), \tag{9}$$

where STCN is the proposed Spatial and Temporal Contexts Network, and $Concat$ is the concatenation operation, $\otimes$ denotes element-wise multiplication.

A comparison of the feature maps extracted before STCN and after STCN is shown in Fig. 3. The first row shows the captured historical frames, the second row shows the feature maps extracted by the feature extraction network, and the third row shows the feature maps after the fusion of spatial-temporal information by STCN. We can observe that features incorporating spatial-temporal contextual information are more focused on the target and can still focus on the target location in the presence of partial occlusion (e.g., frame #5). Since STCN incorporates the spatial-temporal context information of the history frames, the tracker is still able to estimate the target location when the target appears to be completely occluded in the search frame (e.g., the search image and response map on the right side). Relying on the STCN, the spatial-temporal contexts are effectively exploited to refine the feature maps.

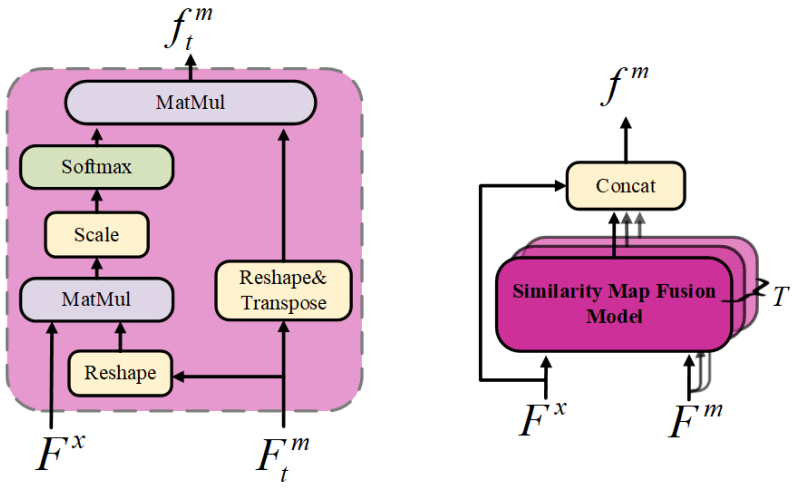
3.4. Multi-Similarity Map Fusion Module

Inspired by [34, 15], we first utilize the embedded Gaussian function to compute the similarity map $W_t \in \mathbb{R}^{HW \times HW}$ between each pixel of F_t^m and F^x:

$$W_{t,ij} = \frac{exp[(F_{t,i\cdot}^m \odot F_{\cdot j}^x)/\sqrt{d}]}{\sum_k exp[(F_{t,k\cdot}^m \odot F_{\cdot j}^x)/\sqrt{d}]}, \tag{10}$$

where i is the index of each pixel of $F_t^m \in \mathbb{R}^{HW \times C}$, j is the index of each pixel of $F^x \in \mathbb{R}^{C \times HW}$, and the operator $\odot$ denotes dot-product; $\frac{1}{\sqrt{d}}$ is the scaling factor similar to [35], and we set d to C, which is the channel of F_t^m. Meanwhile, we normalize W_t with a $softmax$ function. Based on similarity weight map W_t, the t-th fusion information f_t^m can be computed as:

$$f_t^m = (F_t^m)^T \otimes W_t, t = 2, 3, \cdots, T, \tag{11}$$



(a) Similarity Map Fusion Module (b) Multi-Similarity Map Fusion Module

Figure 4: The similarity map fusion module (left) and multi-Similarity map fusion module (right). Here $F^x \in \mathbb{R}^{C\times H\times W}$ and $F^m = \{F_t^m, F_{t+1}^m, \cdots, F_{t+T}^m\}$, $F_t^m \in \mathbb{R}^{C\times H\times W}$, the start frame is t, and the number of memory frames is T, C, H, and W denotes the feature channels, the height, and the width of the feature map, respectively. The reshape operator is to transform the shape of F_t^m from $C \times H \times W$ to $HW \times C$ as well as the shape of F^x from $C \times H \times W$ to $C \times HW$ in order to the convenience of computing. The "MatMul" operator represents matrix multiplication, while the "Concat" operator represents concatenation along the channel dimension.

where $(F_t^m)^T \in \mathbb{R}^{C\times HW}$ is the transpose of F_t^m and $\otimes$ represents the element-wise multiplication. The target information stored in F_t^m is searched adaptively according to the information of F^x.

To allow F^x to retrieve historical target information from different times jointly, we propose the Multi-SMFM. Specifically, we concatenate the fusion information $f_t^m, (t = 2, 3, \cdots, T)$ and the search feature map F^x along the channel dimension to generate the final output feature map f^m, which can be denoted as:

$$f^m = Concat(F^x, f_2^m, f_3^m, \cdots, f_T^m), \tag{12}$$

where $(F_t^m)^T \in \mathbb{R}^{C\times HW}$ is the transpose of F_t^m, and the $Concat$ symbolizes the operation of concatenation, $\otimes$ represents the element-wise product. Relying on the Multi-SMFM, the spatial and temporal contexts are successfully utilized to construct and enhance the similarity map in order to improve

robustness and precision. A diagram of the Multi-SMFM is shown in Fig. 4.

4. Head Network and Loss Functions

The anchor-free trackers [10, 8] have achieved superior performance with lighter model weights than their anchor-based counterparts. Therefore, we employ an anchor-free head network to classify the target from backgrounds and predict the target bounding box.

The anchor-free head network in this work can be divided into a classification branch and a regression branch. Specifically, we encode f^m with a classification convolutional network to achieve the classification task. In addition, we observed that positive samples near the target's border tended to have low-quality bounding box predictions. In order to suppress such classification errors, a sub-branch is forked to construct a center-ness response map. For the regression branch, a regression output is generated from f^m via a lightweight regression convolutional network and used for target bounding box prediction.

As illustrated in Fig. 1, the tracking head network is composed of a classification task that predicts the category at each position and a regression task that compute the target bounding box for each location. For a response map f^m, the classification branch outputs a classification feature map $A^{cls}_{w\times h\times 2}$, which indicates the foreground and background prediction scores for each point $(i, j, :)$ in the search region; the regression branch outputs a regression feature map $A^{reg}_{w\times h\times 4}$, which contains a 4D vector (l, t, r, b) representing the distances from the corresponding location to the four sides of the bounding box in the input search region.

For classification inspired by [10], we apply the cross-entropy loss. For regression, we denote the left-top and right-bottom corners of the ground truth bounding box as (x_0, y_0, x_1, y_1), and denote the mapping point of (i, j) on the search image as (x, y). The labels of regression at $A^{reg}_{w\times h\times 4}$ can be formulated as:

$$\begin{aligned} l &= x - x_0 \quad t = y - y_0 \\ r &= x_1 - x \quad b = y_1 - y. \end{aligned} \tag{13}$$

The regression loss is computed on the positive samples and defined as:

$$L_{reg} = \frac{1}{N_{pos}} \sum_{i,j} L_{IOU}\left(A^{reg}_{w\times h\times 4}(i, j), \mathbb{I}(i, j)\right), \tag{14}$$

where N_{pos} represents the number of positive samples, $\mathbb{I}(i,j)$ is the regression label, and $A^{reg}_{w\times h\times 4}(i,j)$ is the prediction value of regression. L_{IOU} is the Intersection over Union (IOU) between the ground truth and predicted bounding boxes.

In addition, following [36], we add a center-ness branch in parallel with the classification branch to eliminate outliers. The output of the center-ness branch is denoted as $A^{cen}_{w\times h\times 1}$, where each point value corresponds to a position's center-ness score. The center-ness loss is formulated as:

$$\mathcal{L}_{cen} = \frac{-1}{N_{pos}} \sum_{i,j} C(i,j) * \log A^{cen}_{w\times h\times 1}(i,j) + (1 - C(i,j)) * \log\left(1 - A^{cen}_{w\times h\times 1}(i,j)\right), \tag{15}$$

where $C(i,j)$ is the center-ness label defined as the distance between the respective position (x,y) and the object's center inside the search region, which can be computed as:

$$C(i,j) = \frac{\sqrt{\min(l,r) \times \min(t,b)}}{\sqrt{\max(l,r) \times \max(t,b)}}. \tag{16}$$

Note that $C(i,j)$ is set to 0 if (x,y) is in the background.

The overall training loss function is:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{cen} + \lambda_2 \mathcal{L}_{reg}, \tag{17}$$

where $\mathcal{L}_{cls}$ is the entropy loss for classification. The constants λ_1 and λ_2 center-ness weight loss and regression loss, then $\lambda_1 = 1$ and $\lambda_2 = 3$ as in [8].

5. Experiments

In this section, we will first discuss the details of our experimental implementation. Then we evaluate our approach on OTB 2015 [37], GOT-10K [38], TrackingNet [39], LaSOT [40], UAV123 [41], and VOT2018 [42]. Finally, we give an ablation study of the proposed method.

5.1. Implementation Details

5.1.1. Training detials

Training set of TrackingNet [39], COCO [43], GOT-10K [38], ILSVRC DET [44], ILSVRC VID [44] and LaSOT [40] are adopted in the training. We train STCTrack with SGD optimizer for 40 epochs on two NVIDIA RTX

3090 GPU with a batch size of 32 and 300,000 samples per epoch. We sample 3 frames with a maximum gap of 50 frames. The momentum of the optimizer is set to 0.9, and the weight decay rate is set to 1×10^{-4}. We adopt GoogleNet as our backbone φ^z and φ^x. After 1 epoch, the learning rate increases from 1×10^2 to 8×10^2 with a warming technology and drops from 8×10^2 to 1×10^6 with a cosine annealing learning rate plan for the remaining epochs.

5.1.2. Testing details

In the offline testing process, we use a fixed-step strategy to select $t(t = 8)$ history frames as input. At the same time, the target bounding box corresponding to the history frame is generated as a binary matrix, projected into the same space as the history frame by a lightweight convolution, and then added to the corresponding frame as discussed in section 3.2. The first frame is fed to the STCN module as initial prior information. Subsequent history frames are fed sequentially into the STCN module to extract the history information of the target, then fused with the original information to generate the final history features. These features are input to Multi-SMFM together with the search area features. Finally, the head network predicts the classification and regression scores based on the output of the Multi-SMFM.

5.2. Comparsion with the State-of-the-art

5.2.1. OTB2015

OTB2015 [37] is a classical benchmark in visual object tracking. It has 50 challenging videos with substantial variations. It includes 50 challenging videos with significant variations. Lighting variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, and low resolution is all manually tagged into the test sequences. We compare our network with state-of-the-art trackers including SiamAttn [45], STMTrack [15], SparseTT [46], ASRCF [47], Ocean [22], SiamRPN++ [9], SiamCAR [8], ECO [48], SiamGAT [49], PrDiMP-50 [50], TransT [51], SiamFC++ [10], and ATOM [13]. As given Tab. 1, our tracker sets top performance in terms of success and precision and achieves the best precision score of 92.7% and the best success score of 71.9%, as well as outperforms the previous best performance trackers in terms of success (AUC) metric and precision (PR) metric. Compared with these Siamese-based and CF-based trackers (e.g., SiamRPN++, SiamCAR, ASRCF, and ECO), Our tracker achieves remarkable improvements in both precision and success.

Table 1: A performance list on OTB-2015 for a comprehensive comparison of our tracker with competitive trackers published in recent years. The best two results are highlighted in red, and blue, respectively. Trackers are ranked from top to bottom and left to right according to the Precision (PR) values. "**PR**" and "**AUC**" are abbreviations for precision and success, respectively.

Trackers	**PR**	**AUC**
Ours	0.927	0.719
STMTrack [15]	–	0.719
SiamAttn [45]	0.926	0.712
SparseTT [46]	–	0.704
ASRCF [47]	0.922	0.692
Ocean [22]	0.920	0.684
SiamRPN++ [9]	0.914	0.696
SiamCAR [8]	0.910	0.698
ECO [48]	0.910	0.691
SiamGAT [49]	0.907	0.705
PrDiMP-50 [50]	0.903	0.696
TransT [51]	0.899	0.694
SiamFC++ [10]	0.896	0.683
ATOM [13]	0.879	0.667

Our STCTrack module is highly effective in modeling the target's historical data, which sets it apart from other object tracking algorithms that rely on linear or back-propagation update strategies. Unlike Siamese network-based trackers or correlation filtering-based trackers, STCTrack has the ability to adapt to changes in the target's appearance. Our testing indicates that STCTrack outperforms the transformer-based tracker, TransT, with a 2.8% improvement in accuracy and a 2.5% improvement in success metrics. These results demonstrate that our proposed method surpasses most current state-of-the-art trackers, making it an excellent option for short-term, small-scale object tracking.

5.2.2. GOT-10K

GOT-10K [38] is a recently released benchmark for general-purpose object tracking with high diversity. It consists of around 10,000 short videos of real-world moving objects. These sequences were separated into 563 classes of moving objects, 6 tracking attributes, and 87 classes of motion to cover as many complex patterns as feasible in real-world settings. In this study, we

Table 2: A performance list on test split GOT-10k for a comprehensive comparison of our tracker with competitive trackers published in recent years. Average overlap (**A.O.**) and success rates (**S.R.**) at threshold 0.5 and 0.75 are adopted. The best two results are highlighted in red, and blue, respectively.

Tracker	**AO**	**SR**$_{0.5}$	**SR**$_{0.75}$
Ours	0.654	0.758	0.579
STMTrack [15]	0.642	0.737	0.575
KYS [52]	0.636	0.751	0.515
PrDiMP-50 [50]	0.634	0.738	0.543
SiamGAT [49]	0.627	0.743	0.488
RPT [53]	0.624	0.730	0.504
DiMP-50 [12]	0.611	0.717	0.492
D3S [54]	0.597	0.676	0.462
SiamFC++ [10]	0.595	0.695	0.479
Ocean [22]	0.592	0.695	0.479
SiamBAN [7]	0.579	0.684	0.457
SiamCAR [8]	0.569	0.670	0.415
ATOM [13]	0.556	0.634	0.402
SiamRPN++ [9]	0.517	0.616	0.325

use this protocol to train and test our tracker. Tab. 2 compares our proposed tracker to other state-of-the-art trackers in terms of average overlap (A.O.) and success rates (S.R.) at 0.5 and 0.75 thresholds. The existing trackers considered include STMTrack [15], KYS [52], PrDiMP-50 [50], SiamGAT [49], RPT [53], DiMP-50 [12], D3S [54], SiamFC++ [10], Ocean [22], SiamBAN [7], SiamCAR [8], ATOM [13], and SiamRPN++ [9].

As shown in Tab. 2, our tracker outperforms all existing works in terms of A.O. and S.R. at the 0.75 thresholds. Compared with STMTrack, our method improves by 1.2%, 2.1%, 0.4% in the terms of A.O., SR$_{0.5}$, SR$_{0.75}$, respectively, and outperforms the third place tracker KYS [52] by 6.4% for the SR$_{0.75}$ metric. Real-world targets often undergo significant changes in appearance, which can cause traditional object tracking methods to perform poorly. However, our proposed approach effectively utilizes the historical data of the target to address these appearance changes, which is a unique strategy.

Table 3: A performance list on TrackingNet for a comprehensive comparison of our tracker with competitive trackers. Trackers are ranked from top to bottom according to the "**AUC**" values except for our proposed tracker. "**PR**", "**NPR**", and "**AUC**" are abbreviations for precision, normalized precision, and success, respectively. The best two results are highlighted in red, and blue, respectively.

Tracker	**AUC(%)**	**PR(%)**	**NPR(%)**
Ours	81.3	78.5	86.1
STMTrack [15]	80.3	76.7	85.1
DTT [55]	79.6	78.9	85.0
TrDiMP [56]	78.4	73.1	83.3
AutoMatch [57]	76.0	72.5	82.4
PrDiMP-50 [50]	75.8	70.4	81.6
SiamFC++ [10]	75.4	70.5	80.0
SiamAttn [45]	75.2	71.5	81.7
DROL [58]	74.6	70.8	81.7
KYS [52]	74.0	68.8	80.0
DiMP-50 [12]	74.0	68.7	80.1
SiamRPN++ [9]	73.3	69.4	80.0
D3S [54]	72.8	66.4	76.8
ATOM [13]	70.3	64.8	77.1

5.2.3. TrackingNet

TrackingNet [39] is a large-scale tracking dataset that provides a huge number of videos captured outdoors for training and testing purposes. It has 60,643 sequences annotated with more than 14 million dense bounding boxes. It covers 27 different types of objects and 15 tracking attributes. The testing set comprises 511 videos without ground truths. Our tracker is evaluated on the testing set from the dedicated server. As shown in Tab. 3, our tracker obtains the 81.3% in AUC, 78.5% in P.R., and 86.1% in NPR, which outperforms all previous Siamese-based and CF-based state-of-the-art approaches by a large margin. The result demonstrates that our method can achieve state-of-the-art performance in large-scale tracking scenarios.

5.2.4. VOT2018

VOT2018 [42] consists of 60 sequences in 24 item categories. Following the VOT2018 protocol, we compare STCTrack to state-of-the-art trackers, including STMTrack [15], PGNet [20], PrDiMP-50 [50], SiamFC++ [10],

Table 4: A comprehensive comparison of our tracker with competitive trackers on VOT2018. Trackers are ranked according to the EAO scores. The best two results are highlighted in red, and blue, respectively. ↑ indicates the bigger the better, ↓ indicates the lower the better.

Tracker	EAO↑	A↑	R↓
Ours	0.449	0.610	0.154
STMTrack [15]	0.447	0.590	0.159
PGNet [20]	0.447	0.618	0.192
PrDiMP-50 [50]	0.442	0.618	0.165
SiamFC++ [10]	0.426	0.587	0.183
SiamRPN++ [9]	0.414	0.600	0.159
AOTM [13]	0.401	0.590	0.204
UPDT [60]	0.378	0.536	0.182
DaSiamRPN [59]	0.326	0.570	0.337
ECO [48]	0.280	0.484	0.276
SiamRPN [18]	0.244	0.490	0.464

SiamRPN++ [9], AOTM [13], DaSiamRPN [59], SiamRPN [18], ECO [48] and UPDT [60]. The experimental results are presented in Tab. 4, and the EAO ranking outcomes of our STCTrack and other trackers on VOT2018 are shown in Fig. 5.

Our STCTrack has achieved exceptional performance, ranking top in terms of both EAO and robustness(R). In accuracy(A), STCTrack outperforms STMTrack by 2.0%. While our method is slightly less robust than SiamFC++ by 2.9%, it outperforms PrDiMP-50 in terms of robustness by 1.1% and is more suitable for challenging scenarios. However, it's important to note that the accuracy results may not be directly comparable due to the ground truth being a rotating bounding box in the VOT2018 evaluation, while our estimated bounding box is square-shaped.

5.2.5. UAV123

UAV123 [41] is designed for the evaluation of trackers in unmanned aerial vehicle applications. It has more than 110K video frames in 123 video sequences. As shown in Tab. 5, although many objects have quite low resolutions, our proposed STCTrack achieves 65.4% and 85.6% in AUC score and precision, which significantly outperforms recent competitive siamese trackers such as SiamBAN[7], DaSiamRPN[59], and SiamRPN++[9]. The

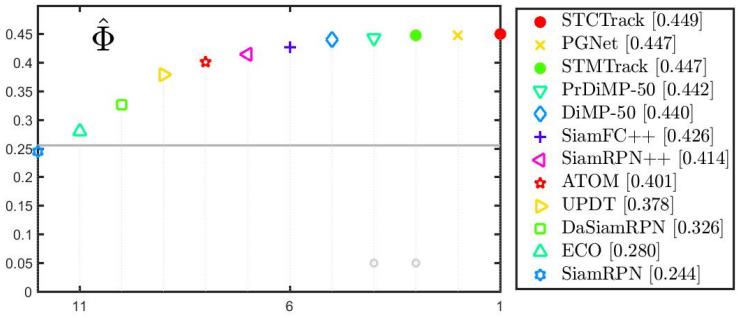


Figure 5: EAO score ranking results on VOT2018.

Table 5: A comprehensive comparison of our tracker with competitive trackers on UAV123 in terms of success (**AUC**) metric and precision metric. The best two results are highlighted in red, and blue, respectively.

Tracker	**AUC**	**Precision**
Ours	0.654	0.856
STMTrack [15]	0.647	–
SiamGAT [49]	0.646	0.843
SiamBAN [7]	0.631	0.833
SiamRPN++ [9]	0.613	0.807
DaSiamRPN [59]	0.586	0.796
ECO [48]	0.537	0.741

results illustrate that our approach is superior for UAV object tracking tasks compared to most existing works.

5.2.6. LaSOT

LaSOT [44] comprises 70 distinct object categories, with each class containing 20 targets. Therefore, the tracker's robustness is essential for these complicated circumstances. We compare STCTrack with the most advanced trackers, including STMTrack [15], DTT [55], PrDiMP-50[50], AutoMatch [57], DiMP-50 [12], SiamRN [61], SiamGAT [49], GlobalTrack [62], SiamBAN [7], SiamCAR [8], ATOM [13] and SiamRPN++ [9]. As shown Tab. 6, Our tracker achieves the highest performance among these trackers, with 63.7% accuracy, 69.6% normalized accuracy, and 61.0% success rate. Compared to the STMTrack, STCTrack achieves a gain of 0.4%, 0.3%, and 0.4% in terms of PR, NPR, and AUC, respectively. Then our method outperforms STM-

Table 6: A comprehensive comparison of our tracker with competitive trackers on LaSOT. Trackers are ranked according to the "**P.R.**" values except for our proposed tracker. "**PR**", "**NPR**" and "**AUC**" are abbreviations for precision, normalized precision, and success, respectively. The best two results are highlighted in red, and blue, respectively.

Tracker	**PR**	**NPR**	**AUC**
Ours	0.637	0.696	0.610
STMTrack [15]	0.633	0.693	0.606
DTT [55]	–	–	0.601
PrDiMP-50[50]	0.609	–	0.598
AutoMatch [57]	0.599	0.675	0.583
DiMP-50 [12]	0.563	0.650	0.569
SiamRN [61]	0.531	–	0.527
SiamGAT [49]	0.530	0.633	0.539
GlobalTrack [62]	0.528	0.597	0.521
SiamBAN [7]	0.521	0.598	0.514
SiamCAR [8]	0.510	0.600	0.507
ATOM [13]	0.505	0.576	0.515
SiamRPN++ [9]	0.491	0.569	0.496

Track by 3.8%, 2.1%, and 2.7% in terms of PR, NPR, and AUC respectively. Due to the incorporation of spatial and temporal contextual information, the accuracy and robustness of the target's feature representation are enhanced.

5.3. Attribution-based Evalution

Furthermore, in order to thoroughly analyze the performance of our STC-Track in various scenarios. We report the results on 12 annotation attributes in LaSOT. Multiple subsets corresponding to attributes that might significantly impact tracking performance are constructed, including Partial Occlusion (POC), Out-of-View (O.V.), Motion Blur (M.B.), Deformation (DEF), Background Clutter (B.C.), Illumination Variation (I.V.), Rotation (ROT), Camera Motion (C.M.), Scale Variation (S.V.), Full Occlusion (FOC), Low Resolution (L.R.) and Aspect Ration Change (ARC).

Fig. 6 depicts the tracking success rate of each tracker for each attribute, with red indicating the best result and green the sub-optimal result. As shown in Fig. 6, compared to existing state-of-the-art trackers, our proposed STCTrack achieves optimal and sub-optimal tracking performance on 10 and 2 attributes, respectively. Since historical spatial and temporal contextual

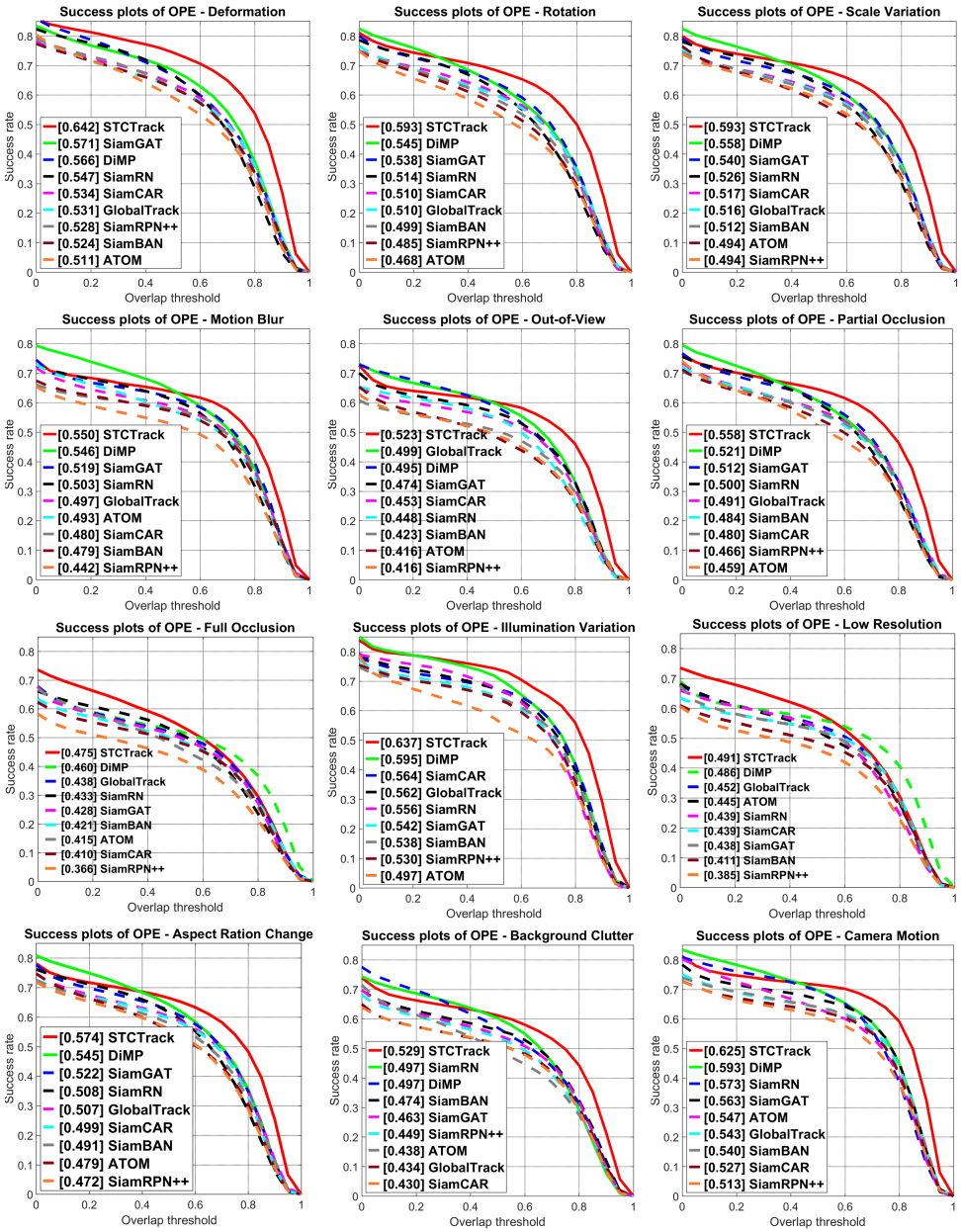


Figure 6: Comparisons on LaSOT with 12 challenging aspects: POC, O.V., MB, DEF, BC, IV, ROT, CM, S.V., FOC, L.R., and ARC. Our STCTrack achieves the best results for 10 attributes.

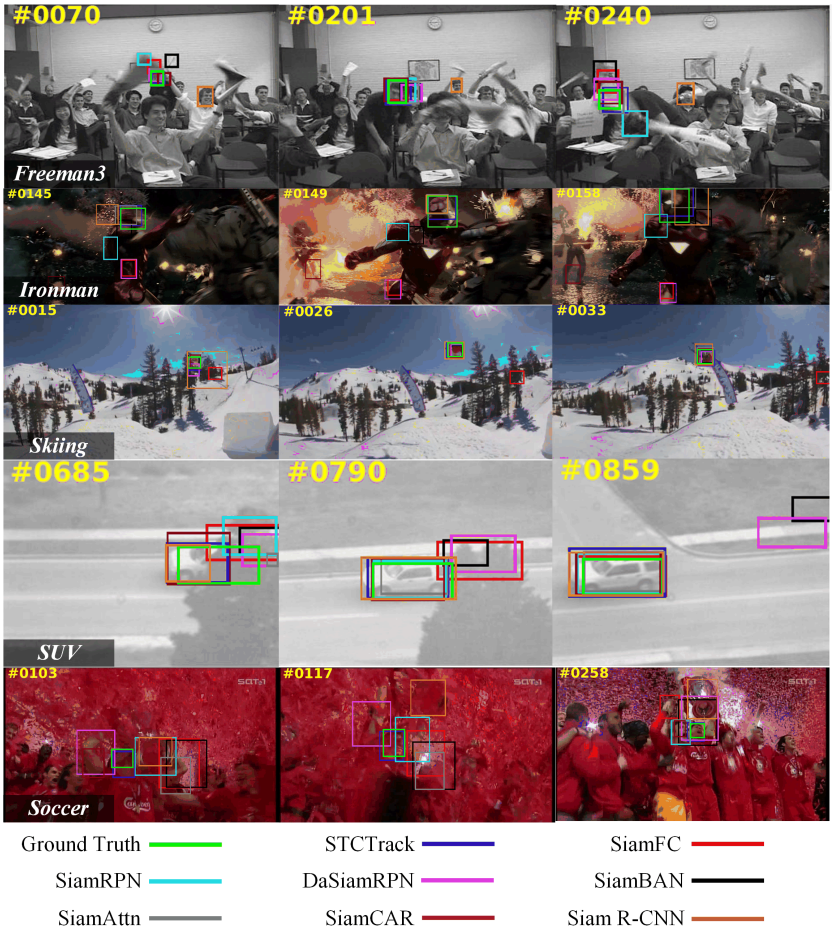


Figure 7: Qualitative results of our STCTrack, along with SiamFC [21], SiamRPN [18], DaSiamRPN [59], SiamBAN [7], SiamAttn [45], SiamCAR [8], Siam R-CNN [63] on five challenge sequences. From top to bottom: *Freeman3*, *Ironman*, *Skiing*, *SUV*, *Soccer*.

information is utilized, the capability of resisting the overall variations in images is learned. Thus an improvement of the performance is achieved by STCTrack across the 10 attributes, e.g., deformation, partial occlusion, and motion blur. The results indicate that STCTrack is better at handling challenging distractors and target appearance changes.

5.4. Qualitative Evaluation

A qualitative comparison is performed with SiamFC [21], SiamRPN [18], DaSiamRPN [59], SiamBAN [7], SiamAttn [45], SiamCAR [8], Siam R-

CNN [63] and proposed method using five challenging images sequences: *Freeman3*, *Ironman*, *Skiing*, *SUV*, *Soccer*. The qualitative comparison is shown in Fig. 7.

We can observe that SiamRPN, SiamFC, SiamBAN, and Siam R-CNN fail to retrieve the target from low resolution and full occlusion in the *Freeman3* experiment. The target in the *Ironman* sequence is affected by deformation, background clutter, illumination variation, and motion blur. Our proposed tracker is still able to perform well compared to other trackers. In the *Skiing*, the resolution of the target is low, and the motion is so fast. The Siam R-CNN fails to track in #15 frame, but our tracker works well. In the *SUV*, the car reappears after being obscured. In the presence of partial and full occlusion, most trackers fail to track the target object, yet our method still tracks the car well. In the *soccer1* sequence, the target face is obscured by background noise and motion blur, and there are a large number of distracting faces with similar features. Only our approach was able to continue to follow the target accurately, as all other methods deviated.

All the above findings suggest that the incorporation of spatial and temporal context information successfully handles the changes in the target's appearance, hence enhancing tracking precision and robustness.

5.5. Ablation Studies

5.5.1. Foreground-background label map

As shown in Tab. 7, we notice that the performance can be improved significantly with foreground-background label maps in the GOT-10K, OTB2015, and LaSOT datasets. This observation suggests that the foreground-background label map can boost performance in the object tracking task.

Table 7: Ablation study on the GOT-10k, OTB2015, and LaSOT. Here the variable "**fb_label**" indicates whether the network should use foreground-background label maps in the memory branch.

	GOT-10K			**OTB2015**		**LaSOT**	
fb_label	**AO**	**SR0.5**	**SR0.75**	**AUC**	**PR**	**AUC**	**PR**
	0.592	0.690	0.509	0.686	0.887	0.560	0.576
✓	**0.654**	**0.758**	**0.579**	**0.719**	**0.927**	**0.610**	**0.624**

5.5.2. The length of history frames

On the GOT-10k dataset, we investigate the performance under different lengths of history frames. As shown in Tab. 8, the proposed STCTacker

Table 8: The performance on GOT-10k with varying numbers of training history frames. The average overlap metric is denoted by **AO**.

#	1	2	3	4	5
AO	0.532	0.638	**0.654**	0.643	0.633

Table 9: The performance of TrackingNet in terms of the success (**AUC**) measure with varying numbers of history frames in the inference phase.

#	2	4	6	8	10	ALL
AUC	69.4	76.2	78.5	**81.3**	78.2	77.0
FPS	40	30	26	20	11	6

achieves the highest performance in terms of average overlap when using 3 historical frames in training.

The length of history frames also has a substantial effect on the performance and running speed in inference. We evaluate the performance and speed with different numbers of historical frames in inference. As shown in Tab. 9, the most appropriate length for this STCTacker is 8. However, increasing the length of frames further does not lead to further performance gain. This is possibly due to the overfitting or too many low-quality frames impacting the tracking process.

5.5.3. Ablation of each component

We conduct an ablation study for both STCN and Multi-SMFM to explore these components' effectiveness. As shown in Tab. 10, compared with the backbone baseline, performance improvement can be achieved by using the STCN (4.6% in A.O., 4.8%, and 7.7% in S.R. at threshold 0.5 and 0.75, respectively). Since the extracted historical features have a large number of redundant target information and some noisy background interference, thus the performance only improves slightly. Likewise, with only multi-SMFM, our tracker improves 3.2% in A.O., 2.5%, and 6.1% in S.R. at thresholds 0.5 and 0.75, respectively. Our tracker yields the best performance with an improvement of about 19.2%, 23.4%, and 35.9% at A.O., S.R. at threshold 0.5 and 0.75, respectively, with STCN and Multi-SMFM.

5.5.4. The number of ConvLSTM layers

In this study, we conducted ablation experiments on the GOT-10K dataset to investigate the impact of the number of ConvLSTM layers and the pres-

Table 10: Ablation study of different parts of our tracker on GOT-10K.

Backbone	STCN	Multi-SMFM	GOT-10K AO	GOT-10K SR0.5	GOT-10K SR0.75
✓			0.532	0.598	0.426
✓	✓		$0.567_{\uparrow 6.6\%}$	$0.637_{\uparrow 6.5\%}$	$0.459_{\uparrow 7.7\%}$
✓		✓	$0.559_{\uparrow 5.1\%}$	$0.623_{\uparrow 4.2\%}$	$0.452_{\uparrow 6.1\%}$
✓	✓	✓	$0.654_{\uparrow 22.9\%}$	$0.758_{\uparrow 26.8\%}$	$0.579_{\uparrow 35.9\%}$

Table 11: Ablation experiments on the number of ConvLSTM layers in the STCN module

layers	skip Con.	GOT-10K AO	GOT-10K SR0.5	GOT-10K SR0.75
{1}	0	0.597	0.679	0.468
{2,2}	0	0.603	0.683	0.478
{2,3,2}	0	0.610	0.695	0.483
	1	0.621	0.703	0.512
{3,3,3}	0	0.617	0.699	0.496
	1	0.630	0.711	0.535
{2,3,3,2}	0	0.622	0.721	0.543
	1	0.641	0.745	0.566
	2	**0.654**	**0.758**	**0.579**
{3,3,3,3}	0	0.593	0.684	0.475
	1	0.643	0.739	0.557
	2	0.649	0.747	0.563

ence of skip connections on tracking performance, and finally determine the optimal structure for the spatio-temporal contextual network. ConvLSTM layers are known to play a crucial role in modeling the appearance of the target, while skip connections are a potent network structural design approach that can enhance network performance and training efficiency.

The tracking performance of different configurations is shown in Tab. 11. As shown in Tab. 11, the STCN module structure with two skip connections and {2, 3, 3, 2} ConvLSTM layers achieves the best tracking performance on the GOT-10K dataset.

Our findings indicate that the STCN module structure with two skip connections and {2, 3, 3, 2} ConvLSTM layers achieves the best tracking performance on the GOT-10K dataset. These results provide valuable insights

into the design of spatio-temporal contextual networks for object tracking and may inform the development of more effective tracking algorithms in the future.

In this study, we conducted ablation experiments on the GOT-10K dataset to explore the optimal spatio-temporal contextual network structure for object tracking. As shown in Tab. 11. Our findings indicate that the STCN module structure with two skip connections and {2, 3, 3, 2} ConvLSTM layers achieves the best tracking performance on the GOT-10K dataset. These results provide valuable insights into the design of spatio-temporal contextual networks for object tracking and may inform the development of more effective tracking algorithms in the future.

6. Conclusion and Discussing

In this paper, we present an end-to-end framework that integrates historical spatial and temporal contexts into object tracking. Our framework includes two new modules: STCN and Multi-SMFM, which are designed to capture historical target information and exploit the target object's features in the current frame to adapt to appearance changes. Our experimental results demonstrate that our proposed tracker outperforms existing state-of-the-art approaches while maintaining a comparable speed of 20 FPS. We hope that our framework will inspire further research on object tracking using spatial and temporal contexts.

One limitation of our study is the absence of a historical frame sampling strategy. In both the training and inference phases, we used a fixed number of historical frames for modeling. However, the data distribution may vary across different datasets, and features may change across different periods of the same video sequence, making it challenging to adopt a non-fixed sampling strategy for adaptive changes. Moreover, we need to filter the history frame samples to eliminate any invalid samples and prevent the introduction of interference information. In future work, we plan to explore historical frame sampling strategies based on our framework.

CRediT authorship contribution statement

Kai Huang: Conceptualization, Methodology, Investigation, Data curation, Writing-Original Draft. **Kai Xiao:** Software, Visualization. **Jun Chu:**

Validation, Project administration, Funding acquisition. **Lu Leng:** Resources, Supervision, Project administration, Funding acquisition. **Xingbo Dong:** Formal analysis, Writing-Review and Editing.

Declaration of Competing Interest

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

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (No. 62162045), the National Natural Science Foundation of China (No. 61866028), Technology Innovation Guidance Program Project (Special Project of Technology Cooperation, Science and Technology Department of Jiangxi Province, No. 20212BDH81003).

References

[1] S. Javed, M. Danelljan, F. S. Khan, M. H. Khan, M. Felsberg, J. Matas, Visual object tracking with discriminative filters and siamese networks: A survey and outlook, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022) 1–20doi:10.1109/TPAMI.2022.3212594.

[2] J. Xing, H. Ai, S. Lao, Multiple human tracking based on multi-view upper-body detection and discriminative learning, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 1698–1701.

[3] Y. Zhang, J. Chu, L. Leng, J. Miao, Mask-refined r-cnn: A network for refining object details in instance segmentation, Sensors 20 (4) (2020) 1010.

[4] H. Cho, Y.-W. Seo, B. V. Kumar, R. R. Rajkumar, A multi-sensor fusion system for moving object detection and tracking in urban driving environments, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 1836–1843.

[5] Y. Yuan, J. Chu, L. Leng, J. Miao, B.-G. Kim, A scale-adaptive object-tracking algorithm with occlusion detection, EURASIP Journal on Image and Video Processing 2020 (2020) 1–15.

[6] J. Chu, Z. Guo, L. Leng, Object detection based on multi-layer convolution feature fusion and online hard example mining, IEEE access 6 (2018) 19959–19967.

[7] Z. Chen, B. Zhong, G. Li, S. Zhang, R. Ji, Siamese box adaptive network for visual tracking, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 6668–6677.

[8] D. Guo, J. Wang, Y. Cui, Z. Wang, S. Chen, Siamcar: Siamese fully convolutional classification and regression for visual tracking, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 6269–6277.

[9] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan, Siamrpn++: Evolution of siamese visual tracking with very deep networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4282–4291.

[10] Y. Xu, Z. Wang, Z. Li, Y. Yuan, G. Yu, Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 12549–12556.

[11] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, F. S. Khan, Learning the model update for siamese trackers, 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 4009–4018.

[12] G. Bhat, M. Danelljan, L. V. Gool, R. Timofte, Learning discriminative model prediction for tracking, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6182–6191.

[13] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, Atom: Accurate tracking by overlap maximization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4660–4669.

[14] T. Yang, A. B. Chan, Learning dynamic memory networks for object tracking, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 152–167.

[15] Z. Fu, Q. Liu, Z. Fu, Y. Wang, Stmtrack: Template-free visual tracking with space-time memory networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13774–13783.

[16] H. Nam, B. Han, Learning multi-domain convolutional neural networks for visual tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4293–4302.

[17] H. Fan, H. Ling, Sanet: Structure-aware network for visual tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 42–49.

[18] B. Li, J. Yan, W. Wu, Z. Zhu, X. Hu, High performance visual tracking with siamese region proposal network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8971–8980.

[19] T. Meinhardt, A. Kirillov, L. Leal-Taixe, C. Feichtenhofer, Trackformer: Multi-object tracking with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8844–8854.

[20] B. Liao, C. Wang, Y. Wang, Y. Wang, J. Yin, Pg-net: Pixel to global matching network for visual tracking, in: European Conference on Computer Vision, Springer, 2020, pp. 429–444.

[21] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. Torr, Fully-convolutional siamese networks for object tracking, in: European conference on computer vision, Springer, 2016, pp. 850–865.

[22] Z. Zhang, H. Peng, J. Fu, B. Li, W. Hu, Ocean: Object-aware anchor-free tracking, in: European Conference on Computer Vision, Springer, 2020, pp. 771–787.

[23] B. Yan, H. Zhao, D. Wang, H. Lu, X. Yang, 'skimming-perusal'tracking: A framework for real-time and robust long-term tracking, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2385–2393.

[24] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, B. Ghanem, Robust visual tracking via consistent low-rank sparse learning, International Journal of Computer Vision 111 (2) (2015) 171–190.

[25] H. Li, Y. Li, F. Porikli, Deeptrack: Learning discriminative feature representations online for robust visual tracking, IEEE Transactions on Image Processing 25 (4) (2015) 1834–1848.

[26] Z. Cui, S. Xiao, J. Feng, S. Yan, Recurrently target-attending tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1449–1458.

[27] D. Gordon, A. Farhadi, D. Fox, Re ̂3: Re al-time recurrent regression networks for visual tracking of generic objects, IEEE Robotics and Automation Letters 3 (2) (2018) 788–795.

[28] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, X. Yang, High-performance long-term tracking with meta-updater, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6298–6307.

[29] S. E. Kahou, V. Michalski, R. Memisevic, C. Pal, P. Vincent, Ratm: recurrent attentive tracking model, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2017, pp. 1613–1622.

[30] T. Yang, A. B. Chan, Recurrent filter learning for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 2010–2019.

[31] Q. Gan, Q. Guo, Z. Zhang, K. Cho, First step toward model-free, anonymous object tracking with recurrent neural networks, arXiv preprint arXiv:1511.06425 (2015).

[32] W. Byeon, Q. Wang, R. K. Srivastava, P. Koumoutsakos, Contextvp: Fully context-aware video prediction, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 753–769.

[33] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, R. S. M. Goh, Anomalynet: An anomaly detection network for video surveillance, IEEE Transactions on Information Forensics and Security 14 (10) (2019) 2537–2550.

[34] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7794–7803.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[36] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.

[37] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: A benchmark, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 2411–2418.

[38] L. Huang, X. Zhao, K. Huang, Got-10k: A large high-diversity benchmark for generic object tracking in the wild, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (5) (2019) 1562–1577.

[39] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, B. Ghanem, Trackingnet: A large-scale dataset and benchmark for object tracking in the wild, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 300–317.

[40] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, H. Ling, Lasot: A high-quality benchmark for large-scale single object tracking, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 5374–5383.

[41] M. Mueller, N. Smith, B. Ghanem, A benchmark and simulator for uav tracking, in: European conference on computer vision, Springer, 2016, pp. 445–461.

[42] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. ˇCehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, et al., The sixth visual object tracking vot2018 challenge results, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 0–0.

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (3) (2015) 211–252.

[45] Y. Yu, Y. Xiong, W. Huang, M. R. Scott, Deformable siamese attention networks for visual object tracking, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 6728–6737.

[46] Z. Fu, Z. Fu, Q. Liu, W. Cai, Y. Wang, Sparsett: Visual tracking with sparse transformers (2022).

[47] K. Dai, D. Wang, H. Lu, C. Sun, J. Li, Visual tracking via adaptive spatially-regularized correlation filters, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4670–4679.

[48] M. Danelljan, G. Bhat, F. Shahbaz Khan, M. Felsberg, Eco: Efficient convolution operators for tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6638–6646.

[49] D. Guo, Y. Shao, Y. Cui, Z. Wang, L. Zhang, C. Shen, Graph attention tracking, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 9543–9552.

[50] M. Danelljan, L. V. Gool, R. Timofte, Probabilistic regression for visual tracking, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 7183–7192.

[51] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, H. Lu, Transformer tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8126–8135.

[52] G. Bhat, M. Danelljan, L. V. Gool, R. Timofte, Know your surroundings: Exploiting scene information for object tracking, in: European Conference on Computer Vision, Springer, 2020, pp. 205–221.

[53] Z. Ma, L. Wang, H. Zhang, W. Lu, J. Yin, Rpt: Learning point set representation for siamese visual tracking, in: European Conference on Computer Vision, Springer, 2020, pp. 653–665.

[54] A. Lukezic, J. Matas, M. Kristan, D3s-a discriminative single shot segmentation tracker, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 7133–7142.

[55] B. Yu, M. Tang, L. Zheng, G. Zhu, J. Wang, H. Feng, X. Feng, H. Lu, High-performance discriminative tracking with transformers, 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 9836–9845.

[56] N. Wang, W. gang Zhou, J. Wang, H. Li, Transformer meets tracker: Exploiting temporal context for robust visual tracking, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 1571–1580.

[57] Z. Zhang, Y. Liu, X. Wang, B. Li, W. Hu, Learn to match: Automatic matching network design for visual tracking, 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 13319–13328.

[58] J. Zhou, P. Wang, H. Sun, Discriminative and robust online learning for siamese visual tracking, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 13017–13024.

[59] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, W. Hu, Distractor-aware siamese networks for visual object tracking, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 101–117.

[60] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, M. Felsberg, Unveiling the power of deep tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 483–498.

[61] S. Cheng, B. Zhong, G. Li, X. Liu, Z. Tang, X. Li, J. Wang, Learning to filter: Siamese relation network for robust tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4421–4431.

[62] L. Huang, X. Zhao, K. Huang, Globaltrack: A simple and strong baseline for long-term tracking, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 11037–11044.

[63] P. Voigtlaender, J. Luiten, P. H. Torr, B. Leibe, Siam r-cnn: Visual tracking by re-detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 6578–6588.