This ICCV Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# i-Siam: Improving Siamese Tracker with Distractors Suppression and Long-Term Strategies

Wei Ren Tan[1] and Shang-Hong Lai[1,2]
[1]National Tsing Hua University, Taiwan
[2]Microsoft AI R&D Center, Taiwan
tanweiren@mx.nthu.edu.tw, lai@cs.nthu.edu.tw

## Abstract

*Recently, Siamese Network (SiamFC) has attracted much attention due to its fast tracking capability. Despite many improvements introduced, its accuracy is still far from human performance. We argue that there are several problems with SiamFC tracker. In particular, the negative signals produced by SiamFC lead to noisy response map. In addition, background noises prevent SiamFC from extracting clean features from the template. To suppress these distractions, first we propose a negative signal suppression approach such that irrelevant features are deactivated. Secondly, we demonstrate that image-level suppression is also important in maximizing the tracking accuracy in addition to the existing feature-level suppression. With better detection sensitivity, we further propose a Diverse Multi-Template (DMT) approach for appearance adaptation while reducing the risk of template drifting during long-term tracking. In our experiments, we conduct extensive ablation studies to demonstrate the effectiveness of the proposed components. Our tracker named* ***i****mproved* ***Siam****ese (****i-Siam****) tracker is able to achieve state-of-the-art results on UAV123, OTB-100, OxUvA, and TLP datasets compared to the existing trackers. Nonetheless, our tracker runs in real time, which is around 43 FPS.*

## 1. Introduction

Automatic visual tracking has been a fundamental and challenging task in the area of artificial intelligence and computer vision [29, 22, 18, 31]. Given only a bounding box of the target without specifying the object type in the first frame, a tracker aims to correctly estimate bounding boxes of the target in subsequent frames. This is useful in many real-world applications, such as video surveillance, autonomous systems, robotics, and so on. However, the performance of recent trackers [26, 17, 32, 10] are still not satisfying due to various challenges, for instance occlusion, background clutter, motion blur, deformation, out-of-view, camera perspective, etc.

Among the existing trackers, Siamese Network (SiamFC) [1] that employed Convolutional Neural Network (CNN) has been a popular tracker explored by many researchers due to its fast tracking speed. Built on SiamFC, many variants [26, 17, 10, 16, 28] have been proposed and achieved better tracking accuracy. However, these SiamFC variants are still far from human performance.

In this work, we inspect the fundamental design of the SiamFC tracker. First, we find that SiamFC employed a convolutional layer as their last layer of the CNN. Hence, the output of CNN also contains negative signals before it is fed to a cross correlation operation as input. Such design is embraced in the traditional signal processing task [30] to find negative correlation between two signals. However, we argue that it is not useful in visual tracking as non-rigid transformation invariance of the target is desired. Furthermore, matched negative signals will also contributes to high confidence scores, resulting in noisy response map. Hence, we propose a **Negative Signal Suppression** approach to deactivate irrelevant features, such that they are exact zero. This is similar to the traditional bag-of-codewords [5] and deep sparse networks [7]. As will be shown in Section 3.2, the proposed approach is able to produce reasonable response map, such that high scores cluster around similar object with less noise.

Second, we also find that the templates used by SiamFC consist of mostly background. Therefore, it can be expected that CNN of SiamFC will extract extremely noisy features from the template. Few works have been proposed to overcome this problem by suppressing irrelevant features via a learned attention module [26] or predefined mask [9]. In this work, we show that it is also important to apply similar technique at *image-level* to maximize the tracking accuracy.

With a more robust detection sensitivity, we further propose Diverse Multi-Template approach to adapt to appearance changes during tracking. Different from the traditional approaches [21, 16], the proposed approach stores a *small*

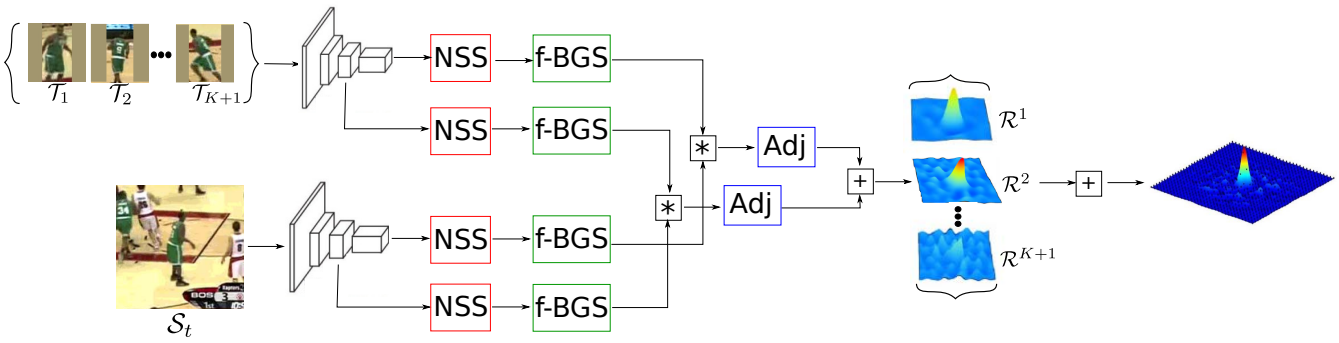



Figure 1: Overview of the proposed i-Siam framework. '*' and '+' are cross-correlation and weighted sum operations, respectively. Background suppression is applied to all templates at image and feature (f-BGS) levels. NSS is the Negative Signal Suppression module. Adj is the adjust layer.

*amount of diverse templates* that are collected during tracking, in addition to the first ground-truth template. Then, multiple confidence maps are computed from each stored templates and fused for a final score map. This prevents the tracker from updating the template model aggressively that causes small error accumulation and template drifting.

As a summary, our contributions in this paper are threefold. First, we propose a Negative Signal Suppression approach to compute less noisy and more robust response map. Secondly, we show that background suppression at image-level coupling with feature-level suppression can improve the tracking accuracy significantly. Finally, we also propose a Diverse Multi-Template approach that is more robust to template drifting during appearance adaptation. In our experiments, extensive ablation studies are conducted in order to demonstrate the effectiveness of each component proposed. The experiments show that our tracker named **i-Siam** achieves state-of-the-art result on UAV123 [20] and OTB-100 [27] datasets, as well as significantly outperforms the existing trackers on two *long-term* tracking datasets, ie. OxUvA [25] and TLP [19]. Nonetheless, the proposed i-Siam tracker is able to perform in real time.

## 2. Related Works

With the great success of deep learning, there emerge a number of deep learning based trackers, which outperformed hand-crafted features [18]. Recently, [1] trained SiamFC tracker using similarity learning. In contrast to candidate proposal based deep trackers [23, 21], SiamFC allows fast computation by matching the target template on a dense grid of a search region in a single evaluation. Due to its competitive performance in real time, many follow-up works were proposed.

EAST [12] employed a cascading approach to early stop the feature extractor when the low-level features are sufficient to track the target in order to speed up the tracker. SINT [23] incorporated optical flow information and achieved better performance at the cost of computation time (4 FPS). CFNet [24] introduced correlation filters (CF) into SiamFC which provide fast solution in Fourier domain. RASNet [26] introduced three kinds of attention mechanisms and achieved better accuracy. SiamRPN [17, 32] merged SiamFC and a region proposal subnetwork to estimate the aspect ratio of the target. SA-Siam [10] utilized complementary appearance and semantic networks to provide richer target representation. Despite the improvements, the tracking performance is still far from human performance.

Few efforts on background suppression were introduced. Traditional correlation filter based trackers [3, 4, 6] introduced spatial regularization in their loss function to reduce background noise. On the other hand, RASNet [26] and SA-Siam [10] adopted careful trained attention modules. Siam-BM [9] showed that predefined mask can achieve similar results without the need of attention modules. However, these works only applied the suppression at feature-level.

Appearance Adaptation is an important ingredient for tracking task. Traditional approaches [2, 21] performed model update by online finetuning the CFs or networks via Stochastic Gradient Descent (SGD). DSiam [8] formulated a closed-form solution in FTT domain for fast update. SiamFC-lu [16] and MemTrack [28] updated their templates via recurrent network. These methods are able to achieve better accuracy but suffer from the model drifting problem.

## 3. Proposed i-Siam Tracker

In this section, we describe our proposed improved Siamese tracker (i-Siam). Our i-Siam tracker is built on the SiamFC tracker [1]. Hence, we first revisit the SiamFC tracking framework. Then, we explain the proposed Negative Signals Suppression and Background Suppression approaches. In addition, we also describe the proposed Diverse Multi-Template approach for appearance adaptation.

Finally, we explain the lost recovery method used in this work for long-term tracking. Figure 1 summarizes the overall framework of the proposed i-Siam tracker.

### 3.1. Overview of the SiamFC Tracker

Assume that a video sequence has $T$ frames, $F_t$ is the frame at time $t \in \{1, \ldots, T\}$. Let $\mathcal{B}_t$ be the bounding box of a target at frame $t$. The bounding box can be described by a four-tuple $(x, y, w, h)$, where $(x, y)$ is the center coordinates and $w$, $h$ are the width and height of the bounding box, respectively. During tracking, a ground-truth bounding box $\mathcal{B}_1$ of a target is given based on $F_1$. The task of a tracker is to predict the bounding boxes $\{\mathcal{B}_2, \ldots, \mathcal{B}_T\}$ for the rest of the sequence $\{F_2, \ldots, F_T\}$.

Let $\mathcal{T}_1$ be the target template image cropped from $F_1$ based on $\mathcal{B}_1$, which is also resized to $127 \times 127$ resolution. On the other hand, the search region $\mathcal{S}_t$ is cropped from $F_t$, centered at previously predicted location based on $\mathcal{B}_{t-1}$. Similarly, $\mathcal{S}_t$ is resized to $255 \times 255$ resolution.

Next, the features $f'_1$ and $f_t$ of $\mathcal{T}_1$ and $\mathcal{S}_t$ are computed by a shared Convolutional Neural Network (CNN), respectively. Formally, this operation can be described as $\Phi : \mathcal{X} \rightarrow \mathcal{F}$, where $\Phi$ is the CNN operation, $\mathcal{T}, \mathcal{S} \in \mathcal{X}$, and $f', f \in \mathcal{F}$. In the SiamFC tracker, $f'_1$ is used as the filter $f^*_t$ of frame $t$ directly, such that $f^*_t = f'_1$. Then, the response map $\mathcal{R}_t$ is calculated with equation 1 via cross correlation operation $(*)$:

$$\mathcal{R}_t = f_t * f^*_t \tag{1}$$

In order to adapt to scale variation, $M$ search regions $\{\mathcal{S}^1_t, \ldots, \mathcal{S}^M_t\}$ with different scales $\{s^1, \ldots, s^M\}$ are cropped from $F_t$. This will produce $M$ response maps $\{\mathcal{R}^1_t, \ldots, \mathcal{R}^M_t\}$, respectively. The response maps will then be upsampled to $272 \times 272$ resolution for more refined localization. The estimation of the center $(x_t, y_t)$ and the scale index $m_t$ for $F_t$ are then determined by maximizing the response maps, i.e.

$$(x_t, y_t, m_t) = \underset{x,y,m}{\arg\max}\, \mathcal{R}^m_t[x, y] \tag{2}$$

Finally, the tracking results are represented as $\mathcal{B}_t = (x_t, y_t, s^{m_t} \cdot w_{t-1}, s^{m_t} \cdot h_{t-1})$. The SiamFC tracker is trained by minimizing the loss function $\mathcal{L}(\mathcal{R}, \hat{y})$ given by

$$\mathcal{L}(\mathcal{R}, \hat{y}) = \mathbb{E}_{\mathcal{Z}}\Big[\frac{1}{|\mathcal{R}|}\sum_u \log(1 + \exp(-\hat{y}[u] \cdot \mathcal{R}[u]))\Big] \tag{3}$$

where $\mathbb{E}_{\mathcal{Z}}$ is the expectation over the training dataset $\mathcal{Z}$. $\hat{y}$ is the ground-truth labels with the same size as the response map $\mathcal{R}$ and $\hat{y}[u] \in \{+1, -1\}$ for each position $u$ in the response map. Meanwhile, $|\mathcal{R}|$ is the number of elements in $\mathcal{R}$. Typically, $\hat{y}[u] = +1$ if $u$ is within a predefined radius of the center, otherwise $\hat{y}[u] = -1$.

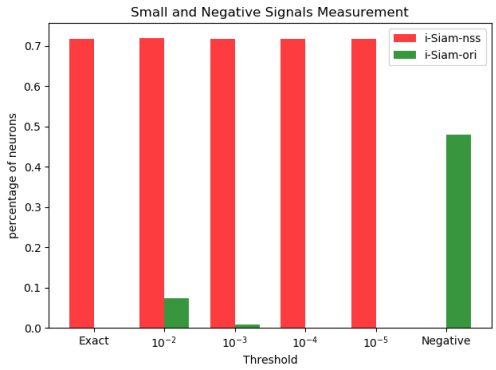



Figure 2: Measures the number of signals that have extremely small or negative value in Siamese trackers with (i-Siam-nss) or without (i-Siam-ori) negative signal suppression on OTB-100 dataset. Exact implies signals that are exactly zero. Negative implies signals with negative value. The rest implies signals with *absolute* value smaller than these thresholds.

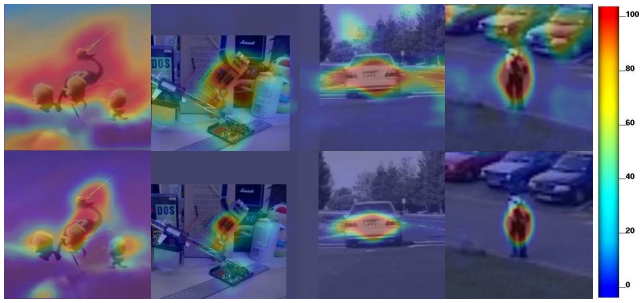



Figure 3: Response maps of dense and sparse i-Siam trackers on OTB-100 dataset. Top: Dense i-Siam tracker (without negative signal suppression). Bottom: Sparse i-Siam tracker (with negative signal suppression). Best viewed in color.

### 3.2. Negative Signal Suppression

After careful inspection of SiamFC [1] tracker's design, we found out that its output has a range of $[-\infty, +\infty]$. As shown in Figure 2, the features extracted by SiamFC tracker (i-Siam-ori) have large amount of negative signals ($\sim 50\%$). Meanwhile, we can also see that most features have strong signal (more than $90\%$ have *absolute* value $> 10^{-2}$), including negative signals.

In the context of traditional signal processing, cross correlation [30] embraces negative signals in order to find negative correlation between two signals. However, we argue that this property is not useful for visual tracking as it is important for the features to be invariant to various transformations, e.g. rotation and deformation. Intuitively, it is more desired to deactivate a feature when it is not related to an object. Such characteristic is embraced by the traditional bag-of-codewords [5] and deep sparse networks [7], which helps learning robust representations.

One may argue that negative signals can be viewed as deactivated neurons. However, these large negative signals will contribute to high confident score of cross correlation when many negative signals match. The evidence is shown in Figure 3 (top) such that the response maps are extremely noisy, with relatively high scores on the background and false positive objects.

To overcome this problem, we propose to suppress the negative signals. In this work, we employed a simple and low computation overhead method, that is Rectified Linear Unit (ReLU) [7] for this purpose:

$$\sigma(x) = \begin{cases} x, & \text{if } x > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where $\sigma$ is the ReLU operation. With this proposed modification, the Siamese tracker can also take advantage of the sparse representation properties. As discussed by [7], sparse representation is robust to small input changes and more linearly separable. Moreover, these properties make optimization easier, which can help the network to learn robust representation more effectively. Similar to [1], logistic function is employed to bound the output of cross correlation to a probabilistic range of $[0, 1]$. However, logistic function requires its input to be in the range of $[-\infty, +\infty]$ while the modified model outputs value of range $[0, +\infty]$. Hence, an adjust layer is stacked on top of the cross correlation layer to adjust the range of its output. In this work, we employed batch normalization layer [14] for this purpose. As a result, around half of the neurons are suppressed to be exact zero (see Figure 2). More importantly, the clusters of the high scores are distributed around similar objects more reasonably, as shown in Figure 3 (bottom).

### 3.3. Background Suppression

As shown in Figure 4 (top), templates are full of noisy information as more than half of the area in the images are background. This makes it hard for CNN to extract features that best represent the target. Hence, it is desired to suppress the background. In general, background suppression can be done in two ways, that are feature-level and image-level.

[26, 9] employed feature-level suppression approach by applying mask to highlight features of the target object. Siam-BM [9] demonstrated that the efforts of training object-specific mask (e.g. [26]) can be bypassed by leveraging the bounding box information. We employ similar approach but with more rigorous suppression criteria. Formally, background suppression can be expressed by equation 5:

$$\bar{f}^* = f^* \cdot \mathcal{M} \tag{5}$$

where $\bar{f}^*$ is the filter with background suppressed and '$\cdot$' is the element-wise product. Meanwhile, the mask $\mathcal{M}$ is determined based on the aspect ratio and a predefined threshold $t_r$, as shown in Figure 5. Unlike [9], we also apply mask when the target has aspect ratio of $\max(\frac{h}{w}, \frac{w}{h}) < t_r$. This is because the image still contains large amount of background noises as shown in Figure 4 (top middle).

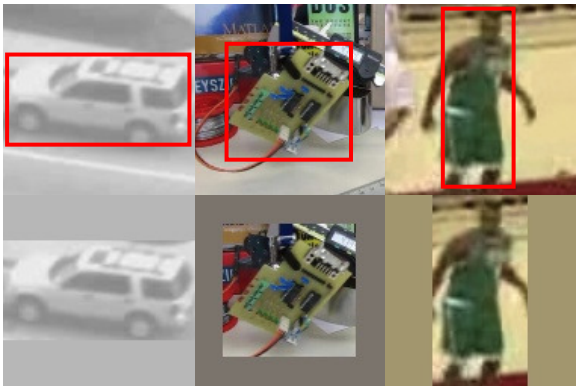


Figure 4: Sample images of image-level background suppression. Objects in the red bounding boxes are the targets. Top: original template. Bottom: after background suppression.

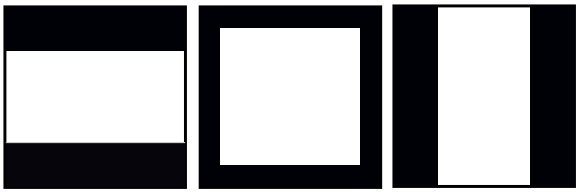


Figure 5: Feature masks when the aspect ratio exceeds a predefined threshold. Left: $\frac{w}{h} > t_r$; Center: $\max(\frac{h}{w}, \frac{w}{h}) < t_r$; Right: $\frac{h}{w} > t_r$.

At image-level, similar suppression approach is applied based on the aspect ratio according to the following equation:

$$\bar{\mathcal{I}} = \mathcal{I} \cdot \mathcal{M} \tag{6}$$

where $\mathcal{I}$ is the original image and $\bar{\mathcal{I}}$ is the image with background suppressed. As shown in Figure 4 (bottom), the values of the suppressed pixels are replaced with the average value of the pixels for each channel.

### 3.4. Diverse Multi-Template approach

One limitation of the SiamFC tracker is its inability to adapt to appearance variations as the template remains unchanged during tracking. Hence, it is desired to employ a template update module into SiamFC. However, existing approaches [2, 21, 8, 16] suffer from template drifting problem. This is because templates obtained during tracking are not always reliable, resulting in small errors accumulation. Interval update was employed in the aforementioned works in order to relax the problem. Despite the efforts, the same

Figure 6: Stored templates (before background suppression) after 200 frames for different sequences of OTB-100. Images in the red bounding box are the first templates for each sequence. The rest are the pseudo-templates.

problem persists and will become worse in long-term tracking task [25].

To overcome this problem, we propose Diverse Multi-Template (DMT) approach. Compared to the aforementioned methods, the key ideas are first, only a small amount of the templates obtained during tracking are stored. Second, these stored templates should be visually diverse. By satisfying these criteria, diverse features can be stored to adapt to appearance changes while avoiding aggressive update.

To be specific, a set of templates $\mathcal{T}^* = \{\mathcal{T}_1, \ldots, \mathcal{T}_{K+1}\}$ are stored in the memory. $K$ is the number of other templates to be stored, in addition to the first template $\mathcal{T}_1$. In this work, we shall name these extra templates as *pseudo-templates* as they are obtained based on the tracker's prediction. Apparently, most of the templates have similar visual appearance. Therefore, it is desired to discard these redundant templates while ensuring the stored templates are diverse.

Given a new candidate template $\mathcal{T}_{K+2}$ obtained from frame $t-1$, we aim to update $\mathcal{T}^*$ based on current $\mathcal{T}^*$ and $\mathcal{T}_{K+2}$. This can be done by computing the distances between these templates in feature space. Then, $\mathcal{T}^*$ will be replaced by $K+1$ templates with largest distances. Note that $\mathcal{T}_1$ is always stored as it is the ground-truth template. Formally, $\mathcal{T}^*$ will be updated if the confidence score $\max(\mathcal{R}_{t-1}) > \tau$, where $\tau$ is a predefined threshold. Let $\{f_1, \ldots, f_{K+2}\}$ be the extracted features for all templates $\{\mathcal{T}_1, \ldots, \mathcal{T}_{K+2}\}$, the distance $d_{ij}$ between any two templates $i, j \in \{1, \ldots, K+2\}$ is then defined as:

$$d_{ij} = ||f_i - f_j||^2 \tag{7}$$

where $||\cdot||$ is the $l_2$-norm. Therefore, the distance $\mathcal{D}_i$ between template $i$ and all other templates can be computed via averaged distance:

$$\mathcal{D}_i = \frac{\sum_{j=1, j\neq i}^{K+2} d_{ij}}{K+1} \tag{8}$$

Since we always use $\mathcal{T}_1$ as one of the templates, we only need to calculate $\{\mathcal{D}_2, \ldots, \mathcal{D}_{K+2}\}$ for $\{\mathcal{T}_2, \ldots, \mathcal{T}_{K+2}\}$. Hence, the templates collection can be updated by replacing $\{\mathcal{T}_2, \ldots, \mathcal{T}_{K+1}\}$ with $K$ pseudo-templates that have larger averaged distances. Figure 6 shows that the proposed DMT approach is able to store diverse templates.

During tracking, all stored templates are used to compute the response maps $\{\mathcal{R}_t^1, \ldots, \mathcal{R}_t^{K+1}\}$ at frame $t$. Intuitively, the fused response map $\mathcal{R}_t$ can be calculated via weighted summation:

$$\mathcal{R}_t = \alpha \mathcal{R}_t^1 + \frac{1-\alpha}{K} \sum_{k=2}^{K+1} \mathcal{R}_t^k \tag{9}$$

where $\alpha$ is the weight to balance the contributions of the first template $\mathcal{R}_t^1$ and other pseudo-templates $\{\mathcal{R}_t^2, \ldots, \mathcal{R}_t^{K+1}\}$.

### 3.5. Lost Recovery for Long-term Tracking

In this paper, we also evaluate our proposed i-Siam tracker on long-term tracking datasets. Hence, it is important to employ a strategy for handling target disappearance and re-appearance problem. In this work, we apply a simply strategy by gradually increasing the search region size to a maximum of $767 \times 767$ resolutions when the detection score is below a predefined threshold. The search region size will be reset to the original size when the target is re-detected successfully.

## 4. Experimental Results

In this section, we evaluate the performance of the proposed i-Siam tracker against the state-of-the-art trackers. We also validate the contribution of each proposed component via ablation studies.

### 4.1. Experiment Settings

We follow most of the settings in [1] in our experiments. Specifically, we employ AlexNet [15] as our Siamese network. All trackers are trained via SGD with momentum of 0.9 on GOT-10K training-set [13]. The weights of the networks are initialized using Xavier method [11]. Training is performed for 50 epochs with mini-batch size of 8. The learning rate is annealed geometrically at each epoch from $10^{-2}$ to $10^{-5}$. To handle scale variations, we search

Table 1: Ablation studies for each proposed component on OTB-100 and TLP. ✗ indicates that the component is removed. NSS: Negative Signals Suppression; f-BGS: Feature-level Background Suppression; I-BGS: Image-level Background Suppression; DMT: Diverse Multi-Template approach

| NSS | f-BGS | I-BGS | DMT | AUC | |
|---|---|---|---|---|---|
| | | | | OTB-100 | UAV123 |
| | | | | **67.7** | **0.586** |
| | | | ✗ | 66.2 | 0.566 |
| | | ✗ | ✗ | 64.5 | 0.564 |
| | ✗ | ✗ | ✗ | 63.3 | 0.554 |
| ✗ | ✗ | ✗ | ✗ | 61.2 | 0.539 |

the target over 3 scales $1.025^{\{-1,0,1\}}$. Empirically, we set $\tau = 0.9$ for templates update and $\alpha = 0.5$ for the response maps fusion.

We conduct our experiments on a machine equipped with one NVIDIA Titan Xp and an Intel Core i7-7700K at 4.20 GHz. With $K = 5$, the proposed i-Siam operates at around 43 FPS, depending on how many times lost recovery is triggered, while still able to achieve competitive performance. The ablation studies are conducted on UAV123 [20] and OTB-100 [27]. Then, the best setting found is employed to validate the performance of the proposed i-Siam on the *long-term* tracking datasets, ie. OxUvA [25] and TLP [19].

### 4.2. Ablation Studies

Table 1 shows the ablation studies on the performance of the proposed tracker when each component is removed. As we can see, the tracking accuracy dropped significantly when each component is removed. By making these simple modifications to address the design flaws of SiamFC tracker, the tracking accuracy can be improved significantly (at least $4.7\%$ AUC increment) with minimal additional computation requirement. In order to understand how $K$ affects the performance, we conducted experiments on OTB-100 using different values for $K$. As shown in Figure 7, $K = 5$ achieved the best result. We also find that setting $K > 5$ does not benefit the tracking accuracy.

### 4.3. Results on OTB-100

OTB-100 [27] is an object tracking dataset with 100 sequences for evaluation. Table 2 reports the performance of the proposed i-Siam and other real-time trackers on OTB-100 based on various attributes. First, we can see that i-Siam is able to achieve top-3 performance on all attributes. Unlike DaSiamRPN [32], i-Siam does not employ aspect ratio adaptation. Furthermore, i-Siam only employs one network while SA-Siam [10] requires two different networks to operate. However, i-Siam is still able to outperform these trackers and achieve the best overall performance. Figure 8 compares i-Siam with the state-of-the-art trackers. i-Siam

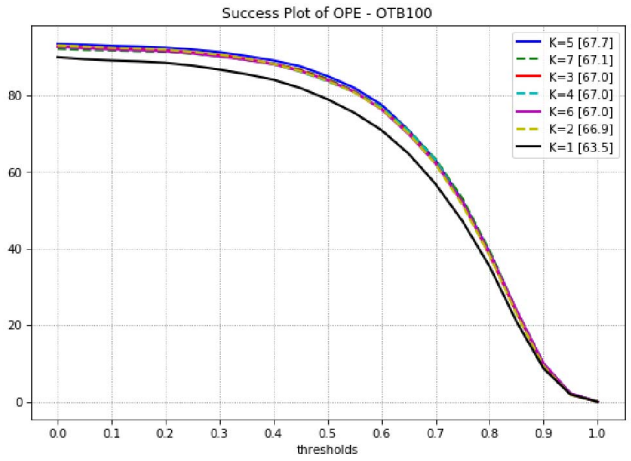



Figure 7: Ablation studies for different numbers of pseudo-templates $K$.

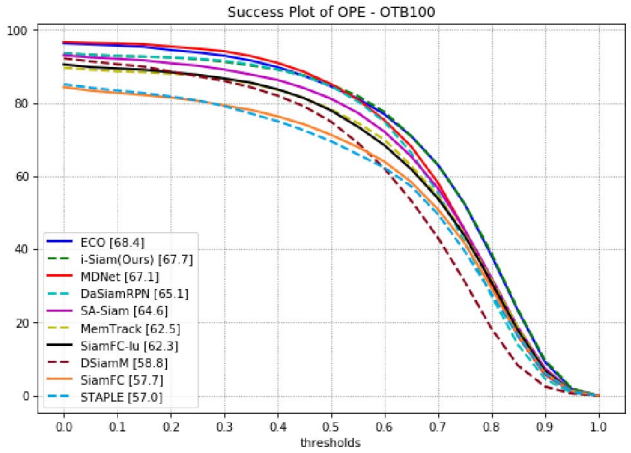



Figure 8: Comparisons between the proposed i-Siam and state-of-the-art trackers on OTB-100 dataset.

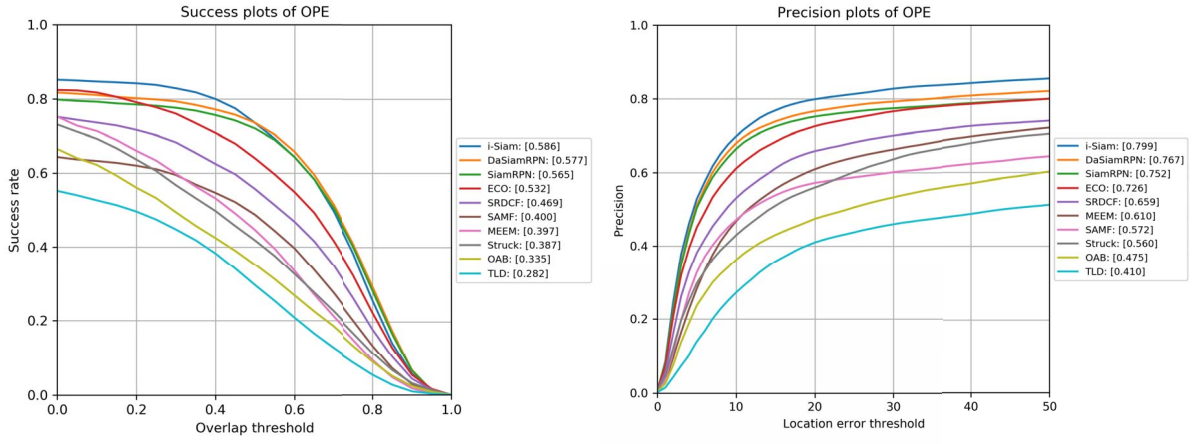
outperformed most of the other trackers including MDNet [21], which is one of the top performing tracker that operates at 1 FPS. While ECO [2] is still the best tracker, its accuracy is only marginally higher against ours ($0.7\%$) but operates at only 8 FPS.

### 4.4. Results on UAV-123

UAV123 [20] is a dataset with sequences captured from drone for aerial tracking. It contains 123 sequences with average sequence length of 915 frames. Figure 9 compares our proposed i-Siam tracker with state-of-the-art trackers. The figure shows that i-Siam tracker is able to outperform other trackers on aerial tracking. These trackers include ECO [2], which is the best performing tracker on OTB-100. In addition, our i-Siam tracker also outperformed DaSiamRPN [32] without aspect ratio adaptation implemented

Table 2: Comparisons between the proposed i-Siam and other real-time trackers based on different attributes for the OTB-100 dataset. The colors refer to the top-3 best performing trackers on each attribute in the order of 1st, 2nd, and 3rd.

| models | background clutter | deformation | fast motion | in-plane rotation | illumination variation | low Resolution | motion blur | occlusion | out-of-plane rotation | Out of view | scale variation | overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **i-Siam (Ours)** | 65.3 | 62.5 | 66.3 | 63.5 | 66.5 | 62.7 | 69.5 | 64.0 | 66.2 | 62.8 | 66.7 | 67.7 |
| DaSiamRPN | 61.6 | 62.6 | 61.6 | 65.5 | 62.2 | 56.0 | 62.9 | 60.0 | 63.9 | 53.8 | 62.9 | 64.7 |
| SA-Siam | 61.2 | 57.7 | 63.2 | 62.5 | 61.6 | 67.7 | 65.7 | 62.4 | 64.3 | 60.8 | 63.9 | 64.6 |
| MemTrack | 58.4 | 52.9 | 62.3 | 60.5 | 58.6 | 58.1 | 62.5 | 58.1 | 60.7 | 55.0 | 60.6 | 62.5 |
| SiamFC-lu | 55.9 | 55.0 | 61.8 | 62.4 | 58.9 | 60.2 | 61.4 | 60.3 | 61.9 | 56.9 | 60.5 | 62.4 |
| DSiamM | 55.7 | 51.8 | 57.2 | 58.9 | 55.9 | 55.9 | 56.2 | 56.1 | 59.0 | 49.7 | 55.8 | 58.8 |
| SiamFC | 49.8 | 50.6 | 58.1 | 56.8 | 54.1 | 57.8 | 57.5 | 54.0 | 56.1 | 50.2 | 56.6 | 57.7 |
| STAPLE | 53.4 | 53.8 | 54.0 | 55.2 | 55.5 | 38.4 | 54.8 | 54.1 | 53.5 | 47.4 | 51.8 | 57.0 |




(a) (b)

Figure 9: Comparisons between the proposed i-Siam and other trackers on UAV-123 dataset. (a) Success plot; (b) Precision plot.

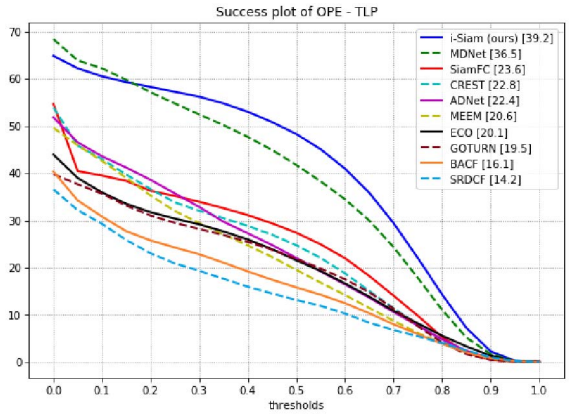



Figure 10: Performance of the proposed i-Siam and existing trackers on TLP dataset.

in our tracker. Based on the fact that i-Siam tracker attained significantly better precision score compared to the second best tracker (3.2% difference as shown in Figure 9(b)), we deduce that i-Siam tracker has significantly better localization capability compared to other trackers.

### 4.5. Results on TLP

TLP [19] is a long-term tracking dataset, consists of 50 videos from real-world scenarios. This dataset encompasses a duration of over 400 minutes or 676K frames. This makes it more than 20 times larger compared to the existing generic tracking datasets in terms of duration per sequence. TLP uses the same evaluation metric as OTB-100. Figure 10 visualizes the performance of the proposed i-Siam and the state-of-the-art trackers. In this experiment, i-Siam tracker significantly outperforms other trackers, including MDNet and ECO. Figure 11 depicts some sample tracking results on TLP.

### 4.6. Results on OxUvA

OxUvA [25] comprises of *dev* and *test* sets with 366 objects in 337 sequences spanning 14 hours. In this work, we compare the proposed i-Siam with other trackers on both sets in open challenge. Unlike other benchmarks, OxUvA uses True Positive Rate (TPR) and True Negative Rate (TNR) to assess the performance. Conceptually, TPR is similar to the measurement used in OTB-100. Meanwhile, TNR reports the ability of a tracker to classify the target as *absent*. To obtained a single measurement, they also

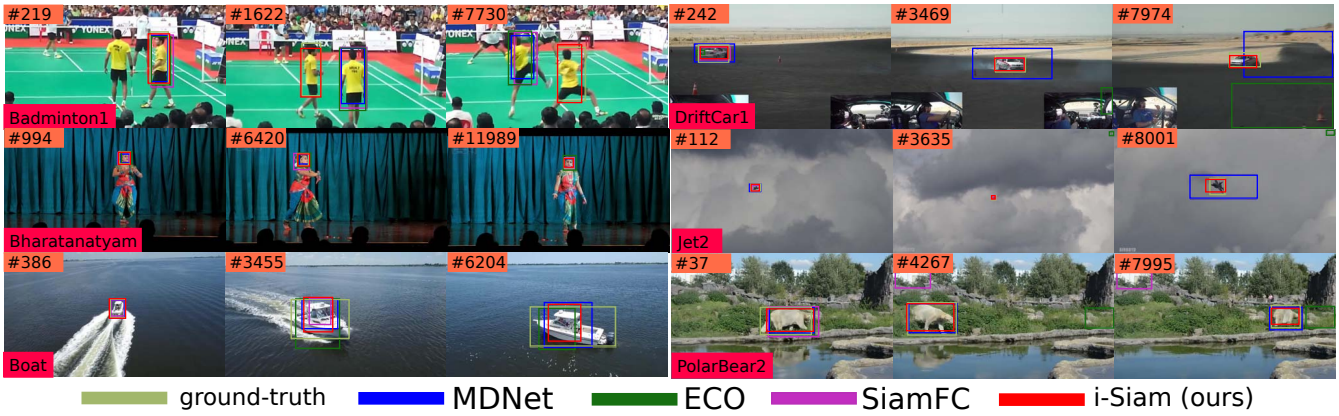
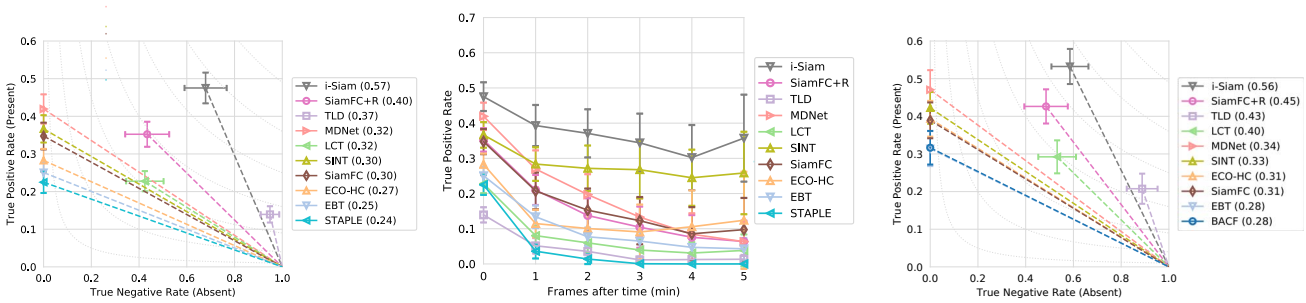



Figure 11: Tracking results of 6 TLP video sequences, using our i-Siam and 3 existing trackers. The images are cropped and enlarged for better visualization. Best viewed in color and pdf.




(a) (b) (c)

Figure 12: Comparisons between the proposed i-Siam and other trackers on OxUvA dataset. (a) Performance on *dev* set; (b) Performance over time on *dev* set; (c) Performance on *test* set.

proposed the maximum geometric mean (MaxGM). However, readers should be noted that MaxGM is also heavily affected by TNR, such that trackers that do not report if a target is *absent* will always have lower MaxGM compared to their counterparts. In this case, comparisons should be done via TPR only. Our i-Siam tracker classifies the target as *absent* if $\mathcal{R}_t < 0.4$.

Figure 12 (a) and (c) shows the performance of the compared trackers on OxUvA. The proposed i-Siam achieved state-of-the-art results on both sets. Meanwhile, Figure 12 (b) shows the performance of the trackers over time. The figure shows that the proposed tracker is able to maintain the performance over time. On the other hand, the performance of the other trackers tend to decay significantly over time. This demonstrates the effectiveness of the proposed DMT on appearance adaptation with better robustness to template drifting.

## 5. Conclusions

In this paper, we showed that existing SiamFC tracker suffers from many design flaws that hampered its performance. In order to overcome these problems, first we proposed a Negative Signal Suppression approach to compute less noisy response map. Furthermore, we demonstrated that combining image-level and feature-level background suppression is able to reduce the noisy information caused by large coverage of background in the template. With better detection sensitivity, we further proposed a Diverse Multi-Template approach for appearance adaptation while maintaining the robustness to template drifting. Our ablation studies showed that these simple modifications are able to improve the tracking accuracy significantly. Finally, we showed that this proposed i-Siam tracker is able to achieve state-of-the-art results on 4 different tracking datasets, including 2 long-term tracking datasets. Nonetheless, it is able to run in real time. For future work, we plan to extend the work to handle disappearance problem and aspect ratio adaptation better.

## References

[1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object

tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 1, 2, 3, 4, 5

[2] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, et al. Eco: Efficient convolution operators for tracking. In *CVPR*, volume 1, page 3, 2017. 2, 4, 6

[3] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 58–66, 2015. 2

[4] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015. 2

[5] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531. IEEE, 2005. 1, 3

[6] H. K. Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, pages 1144–1152, 2017. 2

[7] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011. 1, 3, 4

[8] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang. Learning dynamic siamese network for visual object tracking. In *The IEEE International Conference on Computer Vision (ICCV).(Oct 2017)*, 2017. 2, 4

[9] A. He, C. Luo, X. Tian, and W. Zeng. Towards a better match in siamese network based visual object tracker. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 1, 2, 4

[10] A. He, C. Luo, X. Tian, and W. Zeng. A twofold siamese network for real-time object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4834–4843, 2018. 1, 2, 6

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 5

[12] C. Huang, S. Lucey, and D. Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 105–114, 2017. 2

[13] L. Huang, X. Zhao, and K. Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv preprint arXiv:1810.11981*, 2018. 5

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 4

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 5

[16] B. Li, W. Xie, W. Zeng, and W. Liu. Learning to update for object tracking. *arXiv preprint arXiv:1806.07078*, 2018. 1, 2, 4

[17] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018. 1, 2

[18] P. Li, D. Wang, L. Wang, and H. Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018. 1, 2

[19] A. Moudgil and V. Gandhi. Long-term visual object tracking benchmark. *arXiv preprint arXiv:1712.01358*, 2017. 2, 6, 7

[20] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *European conference on computer vision*, pages 445–461. Springer, 2016. 2, 6

[21] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016. 1, 2, 4, 6

[22] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1442–1468, 2014. 1

[23] R. Tao, E. Gavves, and A. W. Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1420–1429, 2016. 2

[24] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5000–5008. IEEE, 2017. 2

[25] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. Smeulders, P. Torr, and E. Gavves. Long-term tracking in the wild: A benchmark. *arXiv preprint arXiv:1803.09502*, 2018. 2, 5, 6, 7

[26] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4854–4863, 2018. 1, 2, 4

[27] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015. 2, 6

[28] T. Yang and A. B. Chan. Learning dynamic memory networks for object tracking. *arXiv preprint arXiv:1803.07268*, 2018. 1, 2

[29] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006. 1

[30] J.-C. Yoo and T. H. Han. Fast normalized cross-correlation. *Circuits, systems and signal processing*, 28(6):819, 2009. 1, 3

[31] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018. 1

[32] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018. 1, 2, 6