SPARK: Spatial-aware Online Incremental Attack Against Visual Tracking

Qing Guo[1,2*], Xiaofei Xie[2*], Felix Juefei-Xu[3], Lei Ma[4], Zhongguo Li[1], Wanli Xue[5], Wei Feng[1†], and Yang Liu[2]

[1] College of Intelligence and Computing, Tianjin University, China
[2] Nanyang Technological University, Singapore
[3] Alibaba Group, USA
[4] Kyushu University, Japan
[5] Tianjin University of Technology, China
tsingqguo@gmail.com

Abstract. Adversarial attacks of deep neural networks have been intensively studied on image, audio, natural language, patch, and pixel classification tasks. Nevertheless, as a typical while important real-world application, the adversarial attacks of online video object tracking that traces an object's moving trajectory instead of its category are rarely explored. In this paper, we identify a new task for the adversarial attack to visual tracking: online generating imperceptible perturbations that mislead trackers along with an incorrect (Untargeted Attack, UA) or specified trajectory (Targeted Attack, TA). To this end, we first propose a *spatial-aware* basic attack by adapting existing attack methods, *i.e.*, FGSM, BIM, and C&W, and comprehensively analyze the attacking performance. We identify that online object tracking poses two new challenges: 1) it is difficult to generate imperceptible perturbations that can transfer across frames, and 2) real-time trackers require the attack to satisfy a certain level of efficiency. To address these challenges, we further propose the **spatial-aware online incremental attack** (a.k.a. SPARK) that performs spatial-temporal sparse incremental perturbations online and makes the adversarial attack less perceptible. In addition, as an optimization-based method, SPARK quickly converges to very small losses within several iterations by considering historical incremental perturbations, making it much more efficient than basic attacks. The in-depth evaluation of state-of-the-art trackers (*i.e.*, SiamRPN++ with AlexNet, MobileNetv2, and ResNet-50, and SiamDW) on OTB100, VOT2018, UAV123, and LaSOT demonstrates the effectiveness and transferability of SPARK in misleading the trackers under both UA and TA with minor perturbations.

Keywords: Online incremental attack, Visual object tracking, Adversarial attack

1 Introduction

While deep learning achieves tremendous success over the past decade, the recently intensive investigation on image processing tasks *e.g.*, image classification [51,17,41],

[*] Qing Guo and Xiaofei Xie contributed equally to this work.

[†] Wei Feng is the corresponding author (wfeng@tju.edu.cn).

object detection [58], and semantic segmentation [39], reveal that the state-of-the-art deep neural networks (DNNs) are still vulnerable from adversarial examples. The minor perturbations on an image, although often imperceptible by human beings, can easily fool a DNN classifier, detector or segmentation analyzer, resulting in incorrect decisions. This leads to great concerns especially when a DNN is applied in the safety- and security-critical scenarios. For a particular task, the domain-specific study and the understanding of how adversarial attacks influence a DNN's performance would be a key to reduce such impacts towards further robustness enhancement [55].

Besides image processing tasks, recent studies also emerge to investigate the adversarial attacks to other diverse types of tasks, *e.g.*, speech recognition [4,46,6], natural language processing [26,47,60], continuous states in reinforcement learning [49], action recognition and object detection [55,54]. Visual object tracking (VOT), which performs online object localization and moving trajectory identification, is a typical while important component in many safety- and security-critical applications, with urgent industrial demands, *e.g.*, autonomous driving, video surveillance, general-purpose cyber-physical systems. For example, a VOT is often embedded into a self-driving car or unmanned aerial vehicle (UAV) as a key perception component, that drives the system to follow a target object (see Fig. 1). Adversarial examples could mislead the car or UAV with incorrect perceptions, causing navigation into dangerous environments and even resulting in severe accidents. Therefore, it is of great importance to perform a comprehensive study of adversarial attacks on visual object tracking. To this date, however, there exist limited studies on the influence of the adversarial attack on VOT relevant tasks, without which the deployed real-world systems would be exposed to high potential safety risks.

Different from image, speech and natural language processing tasks, online object tracking poses several new challenges to the adversarial attack techniques. *First*, compared with existing sequential-input-relevant tasks, *e.g.*, audios [4], natural languages [26] or videos [55] for classification that have access to the complete sequential data, object tracking processes incoming frames one by one in order. When a current frame t is under attack, all the previous frames (*i.e.*, $\{1, 2 \ldots t-1\}$) are already analyzed and cannot be changed. At the same time, the future frames (*i.e.*, $\{t+1, \ldots\}$) are still unavailable and cannot be immediately attacked as well. With limited temporal data segments and the dynamic scene changes, it is even more difficult to generate imperceptible yet effective adversarial perturbations that can transfer over time (*i.e.*, multiple consecutive frames). *In addition*, the object tracking often depends on a target designated object template cropped from the first frame of a video [1,32] for further analysis. The different initially designated object might lead to different tracking analysis, which renders the universal adversarial perturbation [41] often ineffective.

Furthermore, object tracking usually functions at real-time speed. Thus, it requires the attacks to be efficient enough so that the adversarial perturbation of the current frame can be completed before the next frame arrives. Although the gradient descent-based methods (*e.g.*, FGSM [17], BIM [30]) are demonstrated to be effective in attacking the image classifier, they still encounter efficiency issues in fooling the state-of-the-art trackers when multiple frames quickly arrive.

It is also rather expensive for attacking on multiple frames in real-time [55].

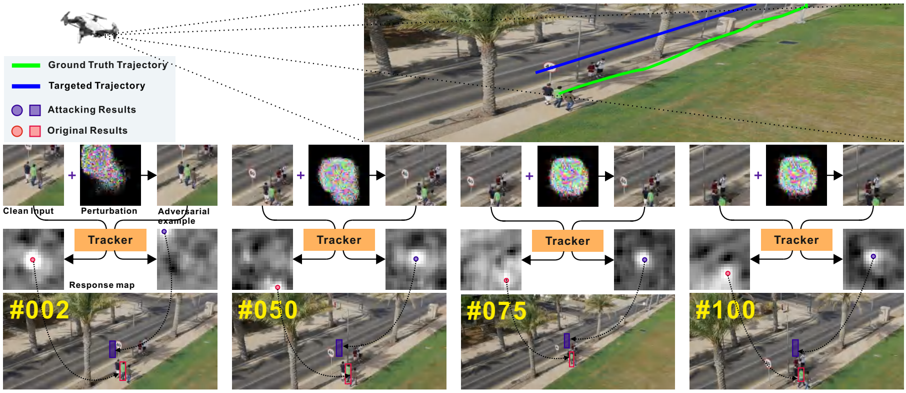

Fig. 1: An example of our adversarial attack to online VOT that drives an UAV [43] to move along the targeted trajectory (the blue line), which causes divergence from the object moving path (the green line). The perturbations are enlarged by ×255 for better visualization.

To better understand the challenges and uniqueness in attacking the VOT, we first propose a *spatial-aware* basic attack method by adapting the existing state-of-the-art attacking techniques (*i.e.*, FGSM, BIM, C&W) that are used to attack each frame individually. Our empirical study confirms that the basic attack is indeed ineffective for attacking the VOT, due to the consecutive temporal frames in real-time. Based on this, we further propose the *spatial-aware online incremental attack* (SPARK) method that can generate more imperceptible perturbations online in terms of both effectiveness and efficiency. The main contributions of this paper are as follows:

- We formalize the adversarial attack problem for the VOT, *i.e.*, generating imperceptible perturbations online to mislead visual trackers that traces an object, into an incorrect (Untargeted Attack, UA) or specified (Targeted Attack, TA) trajectory.
- We propose several *basic attacks* by adapting existing attacks (*i.e.*, FGSM, BIM, C&W) and further perform an empirical study for better understanding challenges of adversarial attacks on real-time object tracking.
- We propose a new *spatial-aware online incremental attack* (SPARK) method that can efficiently generate imperceptible perturbations for real-time VOT.
- In line with the basic methods, our in-depth evaluation demonstrates the effectiveness and efficiency of SPARK in attacking the state-of-the-art SiamRPN++ trackers with AlexNet, MobileNetv2, and ResNet-50 models [32,31] and SiamDW trackers [63] under UA and TA. The generated attacks also exhibit strong transferability to the online updating variants of SiamRPN trackers.

2 Related Work

Adversarial Examples. Extensive studies have shown the vulnerability of DNN from adversarial attacks [35]. [51] initially shown the existence of adversarial attacks, and [17] proposed an efficient one-step method FGSM, that was later improved via iterative method [30] and momentum term [10]. Similarly, [45] proposed the Jacobian-based saliency map attack with high success rate, while [3] realized effective attack by optimization methods (C&W) under different norms. Further adversarial attacks were extended to tasks like object detection [58,33,64], semantic segmentation [58,40], and testing techniques for DNNs [38,59,11].

Recent works also confirmed the existence of adversarial examples in sequential data processing, *e.g.*, speech recognition [6,4,46], natural language [16,26], and video processing [55]. Different from these works, our attack aims at misleading trackers with limited online data access, *i.e.*, the future frames are unavailable, the past frames cannot be attacked either. Among the most relevant work to ours, [55] proposed the $L_{2,1}$ norm-based attack to generate sparse perturbations for action recognition, under the condition that the whole video data is available and the perturbations of multiple frames can be jointly tuned. To further show the difference, we implement a tracking attack with [55] and compare it in the evaluation. [33] attacked the region proposal network (RPN) that is also used in the SiamRPN trackers [32]. However, this attack focuses on fooling image detectors to predict inaccurate bounding boxes, thus cannot be directly used to attack trackers aiming to mislead to an incorrect trajectory with online videos. [54] proposed the video object detection attack by addressing each frame independently, which is not suitable for online tracking where the tracker often runs at real-time speed. Another related work [34] studied when to attack an agent in the reinforcement learning context. In contrast, this work mainly explores how to use temporal constraints to online generate imperceptible and effective perturbations to mislead real-time trackers.

Visual Object Tracking Visual tracking is a fundamental problem in computer vision, estimating positions of an object (specified at the first frame) over frames [57]. The state-of-the-art trackers can be roughly summarized to three categories, including correlation filter-based [8,37,5,61,14,20], classification & updating-based [44,19,48] and Siamese network-based trackers [1,18,65,53,52,13]. Among these works, Siamese network-based methods learn the matching models offline and track objects without updating parameters, which well balances the efficiency and accuracy. In particular, the SiamRPN tracker can adapt objects' aspect ratio changing and run beyond real time [32]. In this paper, we choose SiamRPN++ [31] with AlexNet, MobileNetv2, and ResNet-50 as subject models due to following reasons: 1) SiamRPN++ trackers are widely adopted with high potential to real-world applications [27,31]. The study of attacking to improve their robustness is crucial for industrial deployment with safety concerns. 2) Compared with other frameworks (*e.g.*, correlation filter-based trackers), SiamRPN is a near end-to-end deep architecture with fewer hyper-parameters, making it more suitable to investigate the attacks. In addition to SiamRPN++, we attack another state-of-the-art tracker, *i.e.*, SiamDW [63], to show the generalization of our method.

Difference to PAT [56]. To the best of our knowledge, until now, there has been a limited study on attacking online object tracking. [56] generated physical adversarial textures (PAT) via white-box attack to let the GOTURN tracker [23] lock on the texture when a tracked object moves in front of it. The main differences between our method and PAT are: **(1)** Their attack objectives are distinctly and totally different. As shown in Fig. 2, PAT is to generate *perceptible texture* and let the GOTURN tracker lock on it while our method is to online produce *imperceptible perturbations* that mislead state-of-the-art trackers, *e.g.*, SiamRPN++ [31], along an incorrect or specified trajectory. **(2)** Different theoretical novelties. PAT is to improve an existing Expectation Over

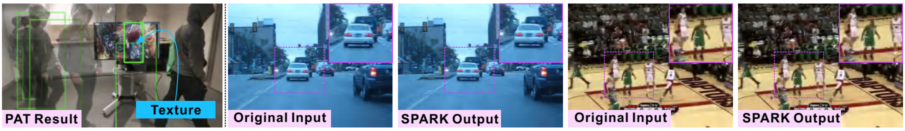

Fig. 2: Difference between PAT [56] and our method. PAT produces *perceptible pattern*.and let the GOTURN tracker lock on the texture. The adversarial perturbations of SPARK are imperceptible and hardly perceived.

Transformation (EOT)-based attack by studying the need to randomize over different transformation variables. Our work intends to perform a comprehensive study on adapt existing adversarial attacks on object tracking and reveal the new challenges in this important task. We then proposed a novel method, *i.e.*, spatial-aware online incremental attack, which can address these challenges properly. **(3)** Different subject models. PAT validates its method by attacking a light deep regression tracker, *i.e.*, GOTURN that has low tracking accuracy on modern benchmarks [12,57,27]. We use our method to attack the state-of-the-art trackers, *e.g.*, SiamRPN++ [31] and SiamDW [63].

3 Spatial-aware Online Adversarial Attack

3.1 Problem Definition

Let $\mathcal{V} = \{\mathbf{X}_t\}_1^T$ be an online video with T frames, where $\mathbf{X}_t$ is the tth frame. Given a tracker $\phi_\theta(\cdot)$ with parameters θ, we crop an object template $\mathbf{T}$ (*i.e.*, the target object) from the first frame. The tracker is tasked to predict bounding boxes that tightly wrap the object in further incoming frames.

To locate the object at frame t, the tracker calculates $\{(y_t^i, \mathbf{b}_t^i)\}_{i=1}^N = \phi_\theta(\mathbf{X}_t, \mathbf{T})$, where $\{\mathbf{b}_t^i \in \Re^{4\times 1}\}_{i=1}^N$ are N object candidates in $\mathbf{X}_t$ and y_t^i indicates the positive activation of the ith candidate (*i.e.*, $\mathbf{b}_t^i$). We denote the tracker's predictive bounding box of the target object at the clean tth frame by $\mathbf{b}_t^{\text{gt}} \in \Re^{4\times 1}$ and the object tracker assigns the predictive result $OT(\mathbf{X}_t, \mathbf{T}) = \mathbf{b}_t^{\text{gt}} = \mathbf{b}_t^k$, where $k = \arg\max_{1\leq i\leq N}(y_t^i)$, *i.e.*, the bounding box with highest activate value is selected as the *predictive object* at frame t. The above tracking process covers most of the state-of-the-art trackers, *e.g.*, Siamese network-based trackers [62,9,13,63,31,21,1] and correlation filter-based trackers [7,50,18]. We define the adversarial attacks on tracking as follows:

Untargeted Attack (UA). UA is to generate adversarial examples $\{\mathbf{X}_t^{\text{a}}\}_1^T$ such that $\forall 1 \leq t \leq T$, $\text{IoU}(OT(\mathbf{X}_t^{\text{a}}, \mathbf{T}), \mathbf{b}_t^{\text{gt}}) = 0$, where $\text{IoU}(\cdot)$ is the Intersection over Union between two bounding boxes.

Targeted Attack (TA). Suppose a *targeted trajectory* $\{\mathbf{p}_t^{\text{tr}}\}_1^T$ desires the trajectory we hope the attacked tracker to output, *e.g.*, the blue line in Fig. 1. TA is to generate adversarial examples $\{\mathbf{X}_t^{\text{a}}\}_1^T$ such that $\forall 1 \leq t \leq T$, $ce(OT(\mathbf{X}_t^{\text{a}}, \mathbf{T})) = \mathbf{p}_t^{\text{tr}}$, where $ce(\cdot)$ shows the center position of the bounding box and $\mathbf{p}_t^{\text{tr}}$ depicts the targeted position at the tth frame.

Intuitively, UA is to make the trackers predict incorrect bounding boxes of a target object at all frames by adding small distortions to online captured frames while TA aims to intentionally drive trackers to output desired object positions specified by the *targeted trajectory*.

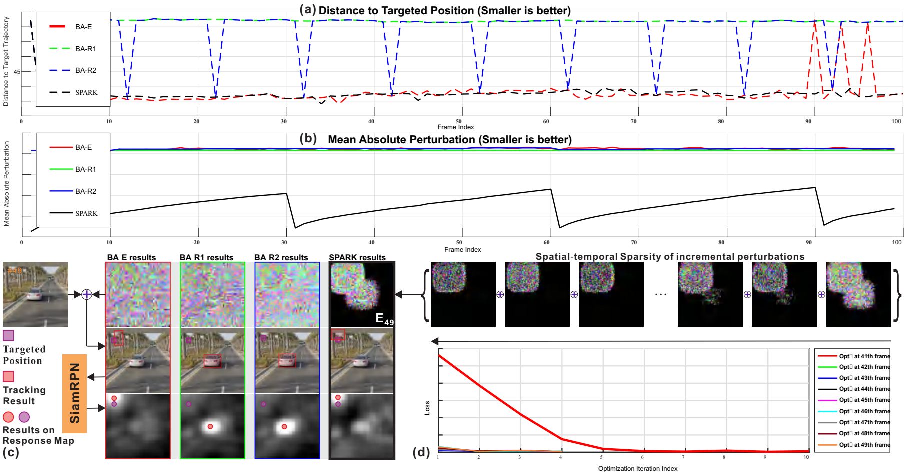

Fig. 3: Analysis of our basic attack (BA) and spatial-aware online incremental attack (SPARK). (a) shows the distance between the targeted position and predicted object position after attacking. A smaller distance means the attack is more effective. (b) shows the mean absolute perturbation of each frame. A smaller MAP leads to less imperceptible perturbation. (c) presents the adversarial perturbations of 4 attack methods at frame 49, corresponding adversarial examples, and response maps from SiamRPN-AlexNet. (d) includes the incremental perturbations from frame 41 to 49 and the loss values at each frame. The perturbations are enlarged by ×255 for better visualization.

3.2 Basic Attack

We first propose the basic attacks by adapting existing adversarial methods at each frame. To attack a tracker $OT(\cdot)$, we can use another tracker $OT'(\cdot)$ to generate adversarial examples. For untargeted attack (UA), at frame t, we formally define the problem of finding an adversarial example as follows:

$$\text{minimize} \;\; \mathcal{D}(\mathbf{X}_t, \mathbf{X}_t + \mathbf{E}_t) \tag{1}$$

$$\text{subject to} \;\; \text{IoU}(OT'(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T}), \mathbf{b}_t^{\text{gt}'}) = 0 \tag{2}$$

where $\mathbf{X}_t^{\text{a}} = \mathbf{X}_t + \mathbf{E}_t$ and $\mathbf{E}_t$ is the desired distortion that changes the result of the tracker and $\mathcal{D}$ is a distance metric. We follow the setup of FGSM and use the L_∞ norm as $\mathcal{D}$. We use $\mathbf{b}_t^{\text{gt}'}$ as the predictive result on the clean frame $\mathbf{X}_t$. When $OT(\cdot) = OT'(\cdot)$, we consider the attack as a white-box attack.

To achieve the UA, we define the objective function f^{ua} such that $\text{IoU}(OT'(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T}), \mathbf{b}^{\text{gt}'}) = 0$ if and only if $f^{\text{ua}}(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T}) < 0$:

$$f^{\text{ua}}(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T}) = y_t^{\text{gt}'} - \max_{\text{IoU}(\text{b}_t^i, \text{b}_t^{gt'})=0} (y_t^i) \tag{3}$$

where $\{(y_t^i, \mathbf{b}_t^i)\}_{i=1}^N = \phi_{\theta'}(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T})$, θ' denotes parameters of $OT'(\cdot)$, and $y_t^{\text{gt}'}$ is the activation value of $\mathbf{b}_t^{gt'}$ at the perturbed frame t. For the targeted attack (TA), at frame t, we define the problem of finding a targeted adversarial example as follows:

$$\text{minimize} \;\; \mathcal{D}(\mathbf{X}_t, \mathbf{X}_t + \mathbf{E}_t) \tag{4}$$

$$\text{subject to} \;\; ce(OT'(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T})) = \mathbf{p}_t^{\text{tr}} \tag{5}$$

Table 1: Comparing basic attacks, *i.e.*, BA-E, BA-R1, and BA-R2 with our SPARK under TA on the OTB100 dataset.

	BA-E			BA-R1			BA-R2			SPARK
	FGSM	BIM	C&W	FGSM	BIM	C&W	FGSM	BIM	C&W	
Succ. Rate (%)	8.0	69.6	57.7	6.6	17.8	17.5	6.7	53.7	23.5	78.9
Mean Absolute Perturbation	1.24	5.88	1.31	1.23	5.96	0.26	1.23	3.36	1.27	1.04
Aver. Iter. Num per frame	1	10	10	0.10	0.95	0.94	0.10	4.6	4.6	2.25
Aver. Cost per frame (ms)	56.2	326.0	264.0	5.50	39.1	24.8	5.68	189.5	121.4	62.1

where $\mathbf{p}_t^{\mathrm{tr}}$ is the targeted position at frame t and $ce(\cdot)$ outputs the center position of a bounding box. To achieve the goal, we define the objective function f^{ta} such that $ce(OT'(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T})) = \mathbf{p}_t^{\mathrm{tr}}$ if and only if $f^{\mathrm{ta}}(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T}) < 0$:

$$f^{\mathrm{ta}}(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T}) = y_t^{\mathrm{gt}'} - \max_{ce(\mathbf{b}_t^i) = \mathbf{p}_t^{\mathrm{tr}}} (y_t^i) \tag{6}$$

To perform the basic tracking attack, FGSM [17], BIM [30] and C&W [3] are adapted to optimize the objective functions (*i.e.*, Eq. (3) and Eq. (6)). In this paper, we mainly focus on the white-box attack on visual object tracking by setting $OT(\cdot) = OT'(\cdot)$ while studying the transferability of different trackers in the experiments.

3.3 Empirical Study

In the following, we perform an empirical study on evaluating the effectiveness of the basic attack. In particular, we study two research questions: 1) how effective is the attack by applying basic attack on each frame? 2) how is its impact of the temporal frames in the video? To answer the questions, we perform two kinds of basic targeted attacks on a state-of-the-art tracker, *i.e.*, SiamRPN-AlexNet[‖]:

BA-E: Online attacking each frame by using FGSM, BIM, and C&W to optimize Eq. (6), respectively.

BA-R: Randomly select some frames and perform the basic attack on these frames using FGSM, BIM, and C&W. For frames between two selected frames, we use the perturbation from the first selected one to distort frames in the interval and see if basic attacks could transfer across time. For example, we attack 1st and 10th frames with basic attacks while distorting the 2th to 9th frames with the perturbation of 1st frame.

Note that BA-E and BA-R can answer the two questions, respectively. To be specific, we have configured two BA-R attacks. First, each frame is selected to be attacked with a probability 0.1 (denoted as **BA-R1**). Second, we perform the basic attack with an interval 10, *i.e.*, attack at the 1th, 11th, 21th, ... frame (denoted as **BA-R2**).

Table 1 shows the success rate, mean absolute perturbation, and average iteration per frame of BA-E, BA-R1, and BA-R2 for attacking SiamRPN-AlexNet-based tracker on OTB100 under TA. We see that: 1) BA-E methods via BIM and C&W get high success rate by attacking each frame. Nevertheless, their perturbations are large and attacking each frame with 10 iterations is time-consuming and beyond real-time tracker. Although FGSM is efficient, its success rate is much lower. 2) Randomly attacking 10% frames, *i.e.*, BA-R1, is about 10 times faster than BA-E. However, the success rate drops

[‖] We select SiamRPN-AlexNet, since it is a representative Siamese network based-tracker and achieves high accuracy on modern tracking benchmarks with beyond real-time speed.

significantly. 3) BA-R2 method attacking at every 10 frames is efficient while sacrificing the success rate. Compared with BA-R1, with the same attacking rate, *i.e.*, 10% frames, BA-R2 has higher success rate than BA-R1. For example, base on BIM, BA-R2 has over two times larger success rate. It infers that perturbations of neighbor 10 frames have some transferability due to the temporal smoothness.

A case study based on BIM is shown in Fig. 3, where we use the three BA attacks to mislead the SiamRPN-AlexNet to locate an interested object at the top left of the scene (targeted position in Fig. 3 (c)). Instead of following the standard tracking pipeline, we crop the frame according to the ground truth and get a region where the object are always at the center. We show the distance between the targeted position (Fig. 3 (a)) and tracking results, and the mean absolute perturbation (MAP) (Fig. 3 (b)) at frame level. We reach consistent conclusion with Table 1. As the simplest solution, BA-E attacks the tracker successfully at some time (distance to the targeted position is less than 20) with the MAP around 5. However, the attack is inefficient and not suitable for real-time tracking. In addition, according to Fig. 3 (c), the perturbations are large and perceptible. The results answer the first question: attacking on each frame is not effective, *i.e.*, time-consuming and bigger MAP.

Consider the temporal property among frames, if the attack can be transferred between the adjacent frames, we could only attack some frames while reducing the overhead, *e.g.*, BA-R1 and BA-R2. Unfortunately, the results in Table 1 and Fig. 3 show that BA-R1 and BA-R2 only work at the specific frames on which the attacks are performed.

The results answer the second question: the perturbations generated by BA is difficult to transfer to the next frames directly due to the dynamic scene in the video (see the results from BA-R1 and BA-R2).

3.4 Online Incremental Attack

Base on the empirical study results from basic attacks, we identify that attacking on each frame directly is not effective. As the frames are sequential and the nearby frames are very similar, our deep analysis found that transferability exists between nearby frames. However, how to effectively use the perturbations from previous frames while being imperceptible when we attack a new coming frame is questionable. A straightforward way is to add previous perturbations to a new calculated one, which will increase the success rate of attacking but lead to significant distortions. To solve this problem, we propose *spatial-aware online incremental attack (SPARK)* that generates more imperceptible adversarial examples more efficiently for tracking. The intuition of SPARK is that we still attack each frame, but apply previous perturbations on the new frame combined with small but effective *incremental perturbation* via optimization.

At frame t, the UA with SPARK is formally defined as:

$$\text{minimize} \;\; \mathcal{D}(\mathbf{X}_t, \mathbf{X}_t + \mathbf{E}_{t-1} + \epsilon_t) \tag{7}$$

$$\text{subject to} \;\; \text{IoU}(OT'(\mathbf{X}_t + \mathbf{E}_{t-1} + \epsilon_t, \mathbf{T}), \mathbf{b}_t^{\text{gt}'}) = 0 \tag{8}$$

where $\mathbf{E}_{t-1}$ is the perturbation of the previous frame (*i.e.*, $t-1$th fame) and ϵ_t is the incremental perturbation. Here, the 'incremental' means $\epsilon_t = \mathbf{E}_t - \mathbf{E}_{t-1}$, and we further have $\mathbf{E}_t = \epsilon_t + \sum_{t_0}^{t-1} \epsilon_\tau$, where $t_0 = t - L$ and $\{\epsilon_\tau\}_{t-L}^{t-1}$ are $L-1$ previous incremental perturbations, and $\epsilon_{t_0} = E_{t_0}$. We denote $t_0 = t - L$ as the start of an attack along the timeline. Based on Eq. 3, we introduce a new objective function by using $L_{2,1}$ norm to regularize $\{\epsilon_\tau\}_{t_0}^{t}$ that leads to small and spatial-temporal sparse ϵ_t.

$$f^{\text{ua}}(\mathbf{X}_t + \epsilon_t + \sum_{t-L}^{t-1} \epsilon_\tau, \mathbf{T}) + \lambda \|\mathit{\Gamma}\|_{2,1}, \tag{9}$$

where $\mathit{\Gamma} = [\epsilon_{t-L}, ..., \epsilon_{t-1}, \epsilon_t]$ is a matrix that concatenates all incremental values.

Similarly, the TA with SPARK is formally defined as:

$$\text{minimize } \ \mathcal{D}(\mathbf{X}_t, \mathbf{X}_t + \mathbf{E}_{t-1} + \epsilon_t) \tag{10}$$

$$\text{subject to } \ ce(OT'(\mathbf{X}_t + \mathbf{E}_{t-1} + \epsilon_t, \mathbf{T})) = \mathbf{p}_t^{\text{tr}}. \tag{11}$$

We also modify the objective function Eq. 6 by adding the $L_{2,1}$ norm and obtain

$$f^{\text{ta}}(\mathbf{X}_t + \epsilon_t + \sum_{t-L}^{t-1} \epsilon_\tau, \mathbf{T}) + \lambda \|\mathit{\Gamma}\|_{2,1}. \tag{12}$$

We use the sign gradient descent to minimize the two objective functions, *i.e.*, Eq. 9 and 12, with the step size of 0.3, followed by a clip operation. In Eq. 9 and 12, λ controls the regularization degree and we set it to a constant 0.00001. Online minimizing Eq. 9 and 12 can be effective and efficient. First, optimizing the incremental perturbation is equivalent to optimizing $\mathbf{E}_t$ by regarding $\mathbf{E}_{t-1}$ as the start point. Since neighboring frames of a video is usually similar, such start point helps get an effective perturbation within very few iterations. Second, the $L_{2,1}$ norm make incremental perturbations to be spatial-temporal sparse and let $\mathbf{E}_t$ to be more imperceptible. For example, when applying SPARK on the SiamRPN-AlexNet-based trackers, we find following observations:

Spatial-temporal sparsity of incremental perturbations: The incremental perturbations become gradually sparse along the space and time (see Fig. 3 (d)). This facilitates generating more imperceptible perturbations than BA methods. In addition, SPARK gets the smallest MAP across all frames with higher success rate than BA-E on OTB100 (see Fig. 3 (b)).

Efficient optimization: Fig. 3 (d) depicts the loss values during optimization from frame 41 to 49. At frame 41, it takes about 7 iterations to converge. However, at other frames, we obtain minimum loss in only two iterations. It enables more efficient attack than BA methods. As presented in Table 1, SPARK only uses 2.25 iterations at average to achieve 78.9% success rate.

The sparsity and efficiency of SPARK potentially avoid high-cost iterations at each frame. In practice, we perform SPARK at every 30 frames** and calculate $\mathbf{E}_{t_0}$ by optimizing Eq. 9 or Eq. 12 with 10 iterations. In addition, we attack on the search region of the attacked tracker instead of the whole frame to accelerate the attacking speed. The search region of the tth frame is cropped from $\mathbf{X}_t$ at the center of predictive result of frame $t-1$, *i.e.*, $\mathbf{b}_{t-1}^{\text{a}}$, and the trackers can be reformulated as $\phi_{\theta'}(\mathbf{X}_t, \mathbf{T}, \mathbf{b}_{t-1}^{\text{a}})$ and $\phi_\theta(\mathbf{X}_t, \mathbf{T}, \mathbf{b}_{t-1}^{\text{a}})$. We will discuss the attack results without $\mathbf{b}_{t-1}^{\text{a}}$ in the experiments.

**We use 30 as the attack interval since videos are usually at 30 fps and such setup naturally utilizes the potential delay between 29th and 30th frames.

We perform both UA and TA against visual tracking and summarize the attack process of SPARK for TA in Algorithm 1. At frame t, we first load a clean frame $\mathbf{X}_t$. If t cannot be evenly divisible by 30, we optimize the objective function, *i.e.*, Eq. 12, with 2 iterations and get ϵ_t. Then, we add ϵ_t into $\mathcal{E}$ that stores previous incremental perturbations, *i.e.*, $\{\epsilon_\tau\}_{t_0}^{t-1}$, and obtain $\mathbf{E}_t = \sum \mathcal{E}$. If t can be evenly divisible by 30, we clear $\mathcal{E}$ and start a new round attack.

```
Algorithm 1: Online adversarial perturbations for TA
Input: A video $\mathcal{V} = \{\mathbf{X}_t\}_1^T$; the object template $\mathbf{T}$;
       targeted trajectory $\{\mathbf{p}_t^{\mathrm{tr}}\}$; the attacked tracker
       $\phi_\theta(\cdot)$; the tracker to perform attack: $\phi_{\theta'}(\cdot)$.
Output: Adversarial Perturbations $\{\mathbf{E}_t\}_1^T$.
Initialize the incremental perturbation set $\mathcal{E}$ as empty;
for $t = 2$ to $T$ do
    Loading frame $\mathbf{X}_t$;
    if $\mathrm{mod}(t, 30) = 0$ then
        max_iter = 10;
        Empty $\mathcal{E}$;
        $t_0 = t$;
    else
        max_iter = 2;
    $\epsilon_t = \mathrm{SPARK}(\phi_{\theta'}(\mathbf{X}_t + \mathcal{E}, \mathbf{T}, \mathbf{b}_{t-1}^{\mathrm{a}}), \mathbf{p}_t^{\mathrm{tr}}, \mathrm{max\_iter})$;
     Add $\epsilon_t$ to $\mathcal{E} = \{\epsilon_\tau\}_{t_0}^{t-1}$;
    $\mathbf{E}_t = \sum \mathcal{E}$;
    $(y_t^{\mathrm{a}}, \mathbf{b}_t^{\mathrm{a}}) = \arg\max_{y_t^i} \phi_\theta(\mathbf{X}_t + \mathbf{E}_t, \mathbf{T}, \mathbf{b}_{t-1}^{\mathrm{a}})$;
    $t = t + 1$;
```

4 Experimental Results

4.1 Setting

Datasets. We select 4 widely used datasets, *i.e.*, **OTB100** [57], **VOT2018** [27], **UAV123** [42], and **LaSOT** [12] as subject datasets. OTB100 and VOT2018 are general datasets that contain 100 videos and 60 videos. UAV123 focuses on videos captured by UAV and includes 123 videos and LaSOT is a large scale dataset containing 280 testing videos.

Models. Siamese network [1,18,32,65,31,13] is a dominant tracking scheme that achieves top accuracy with beyond real-time speed. We select SiamRPN-based trackers [32,31] that use AlexNet [29], MobileNetv2 [25], and ResNet-50 [22] as backbones, since they are built on the same pipeline and achieve the state-of-the-art performance on various benchmarks. We also study the attacks on online udpating variants of SiamRPN-based trackers and the SiamDW tracker [63].

Metrics. We evaluate the effectiveness of adversarial perturbations on the basis of center location error (CLE) between predicted bounding boxes and the ground truth or targeted positions. In particular, given the bounding box annotation at frame t, *i.e.*, $\mathbf{b}_t^{\mathrm{an}}$, we say that a tracker locates an object successfully, if we have $\mathrm{CLE}(\mathbf{b}_t, \mathbf{b}_t^{\mathrm{an}}) = \|ce(\mathbf{b}_t) - ce(\mathbf{b}_t^{\mathrm{an}})\|_2 < 20$ where $\mathbf{b}_t$ is the predicted box [57]. Similarly, we say an attacker succeeds at frame t when $\|ce(\mathbf{b}_t) - \mathbf{p}_t^{\mathrm{tr}}\|_2 < 20$ where $\mathbf{p}_t^{\mathrm{tr}}$ is the tth position on a given targeted trajectory. With above notations, we define precision drop for UA, success rate for TA, and MAP for both UA and TA: (1) **Prec. Drop:** Following [54] and [58], for UA, we use precision drop of a tracker (after attacking) to evaluate the generated adversarial perturbations. The precision of a tracker is the rate of frames where the tracker can locate the object successfully. (2) **Succ. Rate:** For TA, Succ. Rate denotes the rate of frames where an attack method fools a tracker successfully. (3) **MAP:** Following [55], we use the mean absolute perturbation (MAP) to measure the distortion of adversarial perturbations. For a video dataset containing D videos, we have $\mathrm{MAP} = \frac{1}{D*K}\sum_d \sum_k \frac{1}{M*C}\sum_i \sum_c |\mathbf{E}_{k,d}(i,c)|$, where K, M and C refer to the number of frames, pixels and channels, respectively.

Configuration. For TA, the targeted trajectory, *i.e.*, $\{\mathbf{p}_t^{\mathrm{tr}}\}_1^T$, is constructed by adding random offset values to the targeted position of previous frame, *i.e.*, $\mathbf{p}_t^{\mathrm{tr}} = \mathbf{p}_{t-1}^{\mathrm{tr}} + \Delta\mathbf{p}$, where $\Delta \mathbf{p}$ is in the range of 1 to 10. The generated trajectories are often more challenging than manual ones due to their irregular shapes.

4.2 Comparison Results

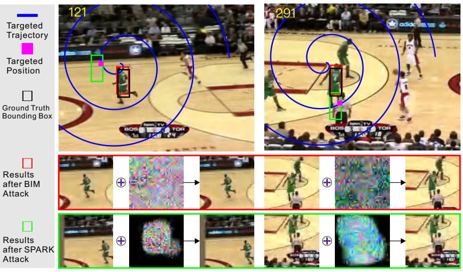

Fig. 4: An TA example based on BIM and SPARK. We use a spiral line as the targeted trajectory that embraces the object at most of the time and makes the TA challenge.

Baselines. Up to present, there still lacks research about adversarial attack on online object tracking. Therefore, we compare with baselines by constructing basic attacks and extending the existing video attack technique. To further demonstrate the advantages of SPARK over existing methods, we extend the BA-E in Table 1 such that it has the same configuration with SPARK for a more fair comparison. To be specific, original BA-E attacks each frame with 10 iterations. However, in Algorithm 1, SPARK attacks every 30 frames with 10 iterations while the frame in interval are attacked with only 2 iterations. We configure the new BA-E with the similar iteration strategy and adopt different optimization methods (*i.e.*, FGSM, BIM [30], MI-FGSM [10], and C&W). In addition, we tried our best to compare with the existing method, *i.e.*, [55] designed for action recognition. However, it uses all frames of a video to predict the category and cannot directly be used for attacking online tracking. We made an extension of it, *i.e.*, when attacking at frame t, the previous 30 frames are used to generate the adversarial.

Results. Table 2 shows the TA/UA results on the four datasets. Column *Org. Prec.* gives the precision of the original tracker. Due to the large evaluation effort, for UAV123 and LaSOT, we only perform the more comprehensive comparison on the smaller model, *i.e.*, SiamRPN-AlexNet.

We observe that: 1) Compared with the existing attacks, SPARK achieves the highest Prec. Drop for UA and Succ. Rate for TA on most of datasets and models. For the results of attacking SiamRPN-Res50 on OTB100, SPARK gets slightly smaller Proc. Drop than MI-FGSM but generates more imperceptible perturbations. 2) SPARK generates imperceptible perturbations. When attacking SiamRPN-AlexNet on all datasets, SPARK always gets more imperceptible perturbations than FGSM, BIM, MI-FGSM, and C&W. [55] produces the smallest perturbations but the attacking is not effective. Similar results can be also found on other three datasets. 3) In general, it is more difficult to attack deeper models for all attacks, since the Prec. Drop and Succ. Rate of almost all attacks gradually become smaller as the models become more complex.

In summary, the results of Table 1 and 2 indicate the effectiveness of SPARK in attacking the tracking models with small distortions. In addition to the quantitative results, we give a concrete example base on BIM and SPARK (see Fig. 4). Compared with BIM, SPARK lets the SiamRPN-AlexNet tracker always produces bounding boxes on the targeted trajectory with a sparse perturbation, indicating the effectiveness of SPARK.

Table 2: Attacking three models with proposed SPARK method on OTB100 and VOT2018 for both UA and TA. The comparison results of 5 existing attack methods are also reported. The results on two larger datasets, *i.e.*, UAV123 and LaSOT, for attacking SiamRPN-AlexNet are presented. The best three results are highlighted by **red**, **green**, and **blue**, respectively.

SiamRPN	Attacks	Untargreted Attack (UA)						Targreted Attack (TA)			
		OTB100			VOT2018			OTB100		VOT2018	
		Org. Prec. (%)	Prec. Drop (%)	MAP	Org. Prec. (%)	Prec. Drop (%)	MAP	Succ. Rate (%)	MAP	Succ. Rate (%)	MAP
AlexNet	FGSM	85.3	8.0	1.24	65.8	13.6	1.24	7.9	1.24	4.3	1.24
	BIM	85.3	**72.1**	2.17	65.8	**57.4**	2.28	**38.8**	2.14	**48.5**	2.10
	MI-FGSM	85.3	**68.4**	3.70	65.8	**58.2**	4.31	**41.8**	3.18	**47.0**	3.17
	C&W	85.3	54.2	1.31	65.8	50.6	1.26	25.7	1.27	25.7	1.23
	Wei	85.3	25.9	0.21	65.8	33.6	0.30	16.0	0.27	20.9	0.24
	SPARK	85.3	**78.9**	1.04	65.8	**61.6**	1.03	**74.6**	1.36	**78.9**	1.38
Mob.	FGSM	86.4	6.7	1.00	69.3	14.1	0.99	7.9	1.00	3.4	0.99
	BIM	86.4	37.8	1.07	69.3	**46.2**	1.06	**30.3**	1.06	**32.9**	1.05
	MI-FGSM	86.4	**42.3**	1.71	69.3	**46.6**	1.73	**33.5**	1.70	**32.7**	1.71
	C&W	86.4	23.6	1.04	69.3	28.2	1.02	13.7	1.05	8.9	1.01
	Wei	86.4	**39.4**	0.84	69.3	27.8	0.54	11.3	0.51	7.0	0.53
	SPARK	86.4	**54.1**	1.66	69.3	**55.5**	1.25	**51.4**	1.65	**45.5**	1.21
Res50	FGSM	87.8	4.5	0.99	72.8	8.1	0.99	7.7	0.92	2.9	0.99
	BIM	87.8	**27.0**	1.10	72.8	**39.1**	1.10	**17.1**	1.09	**17.0**	1.08
	MI-FGSM	87.8	**31.9**	1.72	72.8	**41.8**	1.75	**18.8**	1.71	**19.5**	1.72
	C&W	87.8	14.6	1.03	72.8	20.4	1.01	10.0	1.04	5.3	1.01
	Wei	87.8	9.7	0.65	72.8	15.7	0.68	9.7	0.78	4.8	0.69
	SPARK	87.8	**29.8**	1.67	72.8	**54.3**	1.26	**23.8**	1.70	**39.5**	1.26
SiamRPN	Attacks	Untargreted Attack (UA)						Targreted Attack (TA)			
		UAV123			LaSOT			UAV123		LaSOT	
		Org. Prec.	Prec. Drop	MAP	Org. Prec.	Prec. Drop	MAP	Succ. Rate	MAP	Succ. Rate	MAP
AlexNet	FGSM	76.9	3.7	1.25	43.5	4.0	1.22	3.7	1.25	4.70	1.22
	BIM	76.9	**36.4**	1.70	43.5	**32.0**	1.64	**28.7**	1.75	**17.4**	1.73
	MI-FGSM	76.9	**31.5**	2.54	43.5	**31.6**	2.50	**28.3**	2.53	**17.8**	2.46
	C&W	76.9	17.0	1.37	43.5	19.9	1.29	11.0	1.36	8.7	1.28
	Wei	76.9	5.6	0.31	43.5	9.3	0.29	6.8	0.37	6.9	0.31
	SPARK	76.9	**43.6**	1.13	43.5	**38.2**	0.93	**54.8**	1.06	**48.9**	1.09

4.3 Analysis of SPARK

Validation of the online incremental attack. We implement six variants of SPARK by setting $L \in \{5, 10, 15, 20, 25, 30\}$ in Eq. 12 to analyze how historical incremental perturbations affect attacking results. For example, when attacking the frame t with $L = 5$, we use previous 5 incremental perturbations to generate $\mathbf{E}_t$. We use these SPARKs to attack SiamRPN-AlexNet under TA on OTB100 and report the Succ. Rate, MAP, and MAP difference (MAP Diff.(L)) in Fig. 3, where MAP Diff.(L)=MAP(SPARK(L))-MAP(SPARK($L-1$)). We see that: 1) the Succ. Rate increases with the growing of L. It demonstrates that historical incremental perturbations do help achieve more effective attack. 2) Although MAP also gets larger as the L increases, the MAP Diff. gradually decrease. This validates the advantages of SPARK, that is, it can not only leverage temporal transferability effectively but also maintaining the imperceptible perturbations.

Results under Challenging Attributes. OTB dataset contains 11 subsets corresponding to 11 interference attributes[††]. Fig. 5 shows results of six methods for SiamRPN-AlexNet on 11 subsets. We observe that: 1) SPARK has much larger Prec. Drop and Succ. Rate than baselines on all subsets except the LR one for both UA and TA. 2) The advantages of SPARK over baselines for TA is more significant than that for UA. 3) BIM, Wei, MIFGSM, and C&W are much more effective under the LR attribute than others. This may be caused by the limited effective information in LR frames, which leads to less discriminative deep representation and lets the attacking more easier.

Transferability across Models. We discuss the transferability across models, which is to apply perturbations generated from one model to another. In Table 4, the values in

[††] The 11 attributes are illumination variation (IV), scale variation (SV), in-plane rotation (IPR), outplane rotation (OPR), deformation (DEF), occlusion (OCC), motion blur (MB), fast motion (FM), background clutter (BC), out-of-view (OV), and low resolution (LR).

Table 3: Left sub-table shows the results of attacking DSiamRPN trackers on OTB100 for UA and TA while the right one presents the results of attacking SiamDW trackers.

	UA Attack		TA Attack		UA Attack		TA Attack
	Org. Prec.(%)	Prec. Drop(%)	Succ. Rate(%)		Org. Prec.(%)	Prec. Drop(%)	Succ. Rate(%)
DSiam-AlexNet	86.6	78.5	65.9	SiamDW-CIResNet	83.0	58.1	21.5
DSiam-Mob.	87.8	56.8	44.4	SiamDW-CIResNext	81.7	74.2	29.4
DSiam-Res50	90.3	37.1	20.4	SiamDW-CIResIncep	82.3	70.2	30.8

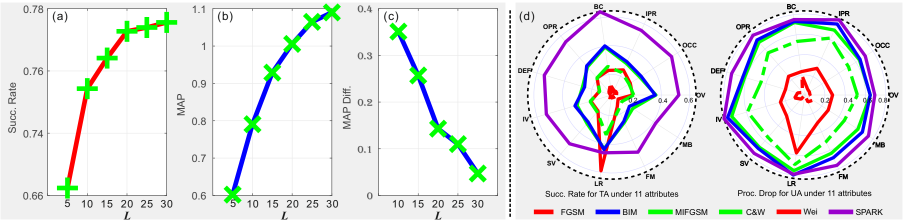

Fig. 5: (a) and (b) are the Succ. Rate and MAP of six variants of SPARK under TA for SiamRPN-AlexNet. The six variants are built by using different number of previous perturbations for Eq. (12) and (c) shows the MAP difference between neighboring variants.(d) Attacking SiamRPN-AlexNet with the six compared methods on the 11 subsets of OTB100 for both TA and UA.

the UA and TA parts are the *Prec. Drop* and *Succ. Rate*, respectively. We see that the transferability across models also exists in attacking object tracking. All attack methods lead to the precision drop to some extent. For example, the perturbations generated by SiamRPN-Res50 cause the precision of SiamRPN-Mob. drop 16.1, which is a huge performance degradation in tracking evaluation. For TA, after transferability, the success rate is around 6.5 for all cases. Such limited transferability may be caused by the insufficient iterations during online process and can be further studied in the future.

SPARK without object template $\mathbf{T}$. As discussed in Section 3.4 and Algorithm 1, the tracked object, *i.e.*, the template $\mathbf{T}$, should be given during attack. Here, we demonstrate that we can realize effective attack without $\mathbf{T}$. Specifically, given the first frame of an online video, we use SSD [36] to detect all possible objects in the frame and select the object nearest to the frame center as the target object. The basic principle behind this is that a tracker usually starts working when the object is within the camera's center view. As presented in Table 4, without the specified $\mathbf{T}$, SPARK-no$\mathbf{T}$ also acheive 71.0% Prec. Drop under UA on OTB100 and is slightly lower than the original SPARK.

SPARK without the attacked tracker's predictions. In Algorithm 1, we detail that our SPARK is performed on the search region of $\phi_{\theta}(\cdot)$ and require the attacked tracker's prediction, *i.e.*, $\mathbf{b}^{\mathrm{a}}_{t-1}$, as an additional input, which might limit the application of our method since we might not access to the attacked tracker's predictions. A simple solution is to replace the $\mathbf{b}^{\mathrm{a}}_{t-1}$ in the Algorithm 1 with $\mathbf{b}^{\mathrm{a}'}_{t-1}$, *i.e.*, we can perform attack on the search region of $\phi_{\theta'}(\cdot)$ and propagate the perturbations to the whole frame. As shown in Table 4, without the attacked tracker's predictions, SPARK-no$\mathbf{b}^{\mathrm{a}}_{t}$ gets 67.7% Prec. Drops under UA on OTB100 which is slightly lower than the original SPARK.

Table 4: The left subtable shows the transferability between subject models (*i.e.*, AlexNet, MobileNetv2, and ResNet50) on OTB100. Values in UA and TA are Proc. Drop and Succ. Rate, respectively. The right subtable shows the results of attacking SiamRPN-AlexNet on OTB100 without object template $\mathbf{T}$ or attacked tracker's prediction $\mathbf{b}_t^{\mathrm{a}}$. The third row of this subtable is the original results of SPARK in Table 2.

	Proc. Drop of UA from			Succ. Rate of TA from				Untargreted Attack (UA)		Targeted Attack (TA)
	AlexNet	Mob.Net	Res50	AlexNet	Mob.Net	Res50		Org. Prec.	Prec. Drop	Succ. Rate
SiamRPN-AlexNet	78.9	6.7	2.0	74.6	6.2	6.7	SPARK-no$\mathbf{T}$	85.3	71.0	50.6
SiamRPN-Mob.	3.5	54.1	2.7	6.3	51.4	6.6	SPARK-no$\mathbf{b}_t^{a}$	85.3	67.7	46.2
SiamRPN-Res50	7.5	16.1	29.8	6.2	6.5	23.8	SPARK	85.3	78.9	74.6

4.4 Attacking other tracking frameworks

Transferability to online updating trackers. We construct three online updating trackers with dynamic Siamese tracking (DSiam) [18], and obtain trackers: DSiamRPN-AlexNet, MobileNetV2, and ResNet-50. We then use the adversarial perturbations from SiamRPN-AlexNet, MobileNetV2, and ResNet-50 to attack the DSiamRPN-based trackers. In Table 3, we observe that: 1) DSiam indeed improves the precision of three SiamRPN-based trackers according to the results in Table 2. 2) The adversarial perturbations from SiamRPNs is still effective for DSaim versions with the precision drops being 78.5%, 56.8%, and 37.1% which are larger than the results in Table 2. This is because DSiamRPN-based trackers use online tracking results that may have been fooled by attacks to update models and make them less effective, thus are easily attacked.

Attacking SiamDW [63]. We validate the generality of SPARK by attacking another tracker, *i.e.*, SiamDW [63] that won the VOT-19 RGB-D challenge and achieved the runner-ups in VOT-19 Long-term and RGB-T challenges [28]. As shown in right sub-table of the Table 3, without changing any attack parameters, SPARK significantly reduces the precision of SiamDW trackers under the UA, demonstrating its generality.

5 Conclusion

In this paper, we explored adversarial perturbations for misleading the online visual object tracking along an incorrect (untarged attack, UA) or specified (targeted attack, TA) trajectory. An optimization-based method, namely *spatial-aware online incremental attack* (SPARK), was proposed to overcome the challenges introduced in this new task. SPARK optimizes perturbations with a $L_{2,1}$ regularization norm and considers the influence of historical attacking results, thus is more effective. Experimental results on OTB100, VOT2018, UAV123, and LaSOT showed that SPARK successfully fool the state-of-the-art trackers.

6 Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61671325, Grant 61572354, Grant 61672376, Grant U1803264, and Grant 61906135, the Singapore National Research Foundation under the National Cybersecurity R&D Program No. NRF2018NCR-NCR005-0001 and the NRF Investigatorship No. NRFI06-2020-0022, and the National Satellite of Excellence in Trustworthy Software System No. NRF2018NCR-NSOE003-0001. We also gratefully acknowledge the support of NVIDIA AI Tech Center (NVAITC) to our research.

7 Supplementary Material

7.1 Attacking Correlation Filter-based Trackers

Correlation filter (CF) is a dominant tracking framework that can achieves well balance between tracking speed and accuracy. However, most of the CF-based trackers are not end-to-end architectures and use hand-craft features. Hence, it is difficult to attack them via the white-box setup and is meaningful to explore if SPARK could attack CF-based trackers by using deep tracking frameworks, *e.g.*, SiamRPN-based trackers. As shown in Table I, the adversarial examples from SiamRPN-Alex can reduce all tested CF-based trackers having different features, which demonstrates that the transiferability of our attack across different trackers and features exists. In terms of different features, the HOG feature is easier attacked when compared with the gray feature, hybird feature (*i.e.*, HOG+CN), and deep feature (*e.g.*, VGG).

Table I: Untargeted attack (UA) for correlation filter-based trackers, *e.g.*, MOSSE [2], KCF [24], BACF [15], STRCF [32], and ECO [8] with the perturbations generated from SiamRPN-AlexNet.

	MOSSE	KCF	BACF	STRCF	ECO
Features	Gray	HOG	HOG+CN	HOG+CN	VGG
Org. Prec. (%)	41.7	69.2	70.5	72.3	89.6
Proc. Drop (%)	0.2	3.3	2.1	1.5	0.9

7.2 Speed Analysis

We have reported the time cost of our SPARK in Table. 1 in the submission and shown that SPARK is more suitable for attacking online trackers than three basic attack methods due to the balance between time cost and attack Succ. Rate. Please find details in Section 3.3. Compared with trackers' cost shown in Table. III, the time cost of our attack method increases as the tracking model becomes larger under the white-box attack. In particular, when attacking SiamRPN-Alex, SPARK achieves near real-time attacking. Although the attack speed decreases with more complex models, the corresponding tracking speed is also slower and lets the influence of decreased attacking be smaller. We can reduce the high time cost of attacking larger models (*e.g.*, MobileNetv2 and Res50) by using the light one (*e.g.*, AlexNet) due to the existence of the transferability between models as discussed in Section 4.3 and Table 4. Specifically, we attack three trackers,

Table II: Time cost of attacks w.r.t. different trackers on OTB100 dataset.

SiamRPN	AlexNet	MobileNetV2	Res50
Track cost per frame (ms)	9.3	37.6	42.1
Attack cost per frame (ms)	41.4	126.9	156.3
Track speed (fps)	108.4	15.3	16.8
Attack speed (fps)	24.3	8.0	6.4

i.e., SiamRPN-Alex/Mob./Res50, via SPARK with the adversarial perturbations generated from SiamRPN-Alex. Then, we calculate the attack's online speed as well as the three trackers' speed. As shown in the following Table. III, the speed of SPARK base on SiamRPN-Alex can reach near real-time speed (around 25 fps) for different trackers, which means our method is suitable for attacking real-time online trackers.

Table III: Time cost of attacking trackers on OTB100. The adversarial perturbations are generated from SiamRPN-Alex.

SiamRPN	AlexNet	MobileNetV2	Res50
Track speed (fps)	108.4	15.3	16.8
Attack speed (fps)	24.3	23.1	22.7

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: arXiv preprint arXiv:1606.09549 (2016)
2. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: CVPR (2010)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57 (2017)
4. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: arXiv:1801.01944 (2018)
5. Chen, Z., Guo, Q., Wan, L., Feng, W.: Background-suppressed correlation filters for visual tracking. In: ICME. pp. 1–6 (2018)
6. Cisse, M., Adi, Y., Neverova, N., Keshet, J.: Houdini: Fooling deep structured prediction models. In: arXiv:1707.05373 (2017)
7. Dai, K., Dong Wang, H.L., Sun, C., Li, J.: Visual tracking via adaptive spatially-regularized correlation filters. In: CVPR. pp. 4665–4674 (2019)
8. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Eco: Efficient convolution operators for tracking. In: CVPR. pp. 6931–6939 (2017)
9. Dong, X., Shen, J.: Triplet loss in siamese network for object tracking. In: ECCV. pp. 472–488 (2018)
10. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: CVPR. pp. 9185–9193 (2018)
11. Du, X., Xie, X., Li, Y., Ma, L., Liu, Y., Zhao, J.: Deepstellar: model-based quantitative analysis of stateful deep learning systems. In: ESEC/FSE. pp. 477–487 (2019)
12. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: CVPR. pp. 5369–5378 (2019)
13. Fan, H., Ling, H.: Siamese cascaded region proposal networks for real-time visual tracking. In: CVPR. pp. 7944–7953 (2019)
14. Feng, W., Han, R., Guo, Q., Zhu, J., Wang, S.: Dynamic saliency-aware regularization for correlation filter-based object tracking. IEEE TIP **28**(7), 3232–3245 (2019)
15. Galoogahi, H.K., Fagg, A., Lucey, S.: Learning background-aware correlation filters for visual tracking. In: ICCV. pp. 1144–1152 (2017)
16. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: SPW. pp. 50–56 (2018)

17. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: arXiv:1412.6572 (2014)
18. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic Siamese network for visual object tracking. In: ICCV. pp. 1781–1789 (2017)
19. Guo, Q., Feng, W., Zhou, C., Pun, C., Wu, B.: Structure-regularized compressive tracking with online data-driven sampling. IEEE TIP **26**(12), 5692–5705 (2017)
20. Guo, Q., Han, R., Feng, W., Chen, Z., Wan, L.: Selective Spatial Regularization by Reinforcement Learned Decision Making for Object Tracking. IEEE TIP **29**, 2999–3013 (2020)
21. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: CVPR. pp. 4834–4843 (2018)
22. He, K., Zhang, X., Ren, S., Sun., J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
23. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: ECCV. pp. 749–765 (2016)
24. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE TPAMI **37**(3), 583–596 (2015)
25. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. In: arXiv preprint arXiv:1704.04861 (2017)
26. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Is bert really robust? natural language attack on text classification and entailment. In: arXiv:1907.11932 (2019)
27. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pfugfelder, R., Zajc, L.C., Vojir, T., Bhat, G., Lukezic, A., Eldesokey, A., Fernandez, G., et al.: The sixth visual object tracking vot2018 challenge results. In: ECCVW. pp. 3–53 (2018)
28. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., Čehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., Eldesokey, A., Kapyla, J., Fernandez, G.: The seventh visual object tracking vot2019 challenge results. In: ICCVW. pp. 2206–2241 (2019)
29. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
30. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. ICLR (Workshop) (2017)
31. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: CVPR. pp. 4282–4291 (2019)
32. Li, B., Wu, W., Zhu, Z., Yan, J., Hu, X.: High performance visual tracking with siamese region proposal network. In: CVPR. pp. 8971–8980 (2018)
33. Li, Y., Tian, D., Chang, M.C., Bian, X., Lyu, S.: Robust adversarial perturbation on deep proposal-based models. In: BMVC. pp. 1–11 (2018)
34. Lin, Y.C., Hong, Z.W., Liao, Y.H., Shi, M.L., Liu, M.Y., Sun, M.: Tactics of adversarial attack on deep reinforcement learning agents. In: IJCAI. pp. 3756–3762 (2017)
35. Ling, X., Ji, S., Zou, J., Wang, J., Wu, C., Li, B., Wang, T.: Deepsec: A uniform platform for security analysis of deep learning model. In: IEEE Symposium on Security and Privacy (SP). pp. 673–690 (2019)
36. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37 (2016)
37. Lukežič, A., Vojíř, T., Čehovin, L., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: CVPR. pp. 4847–4856 (2017)
38. Ma, L., Juefei-Xu, F., Zhang, F., Sun, J., Xue, M., Li, B., Chen, C., Su, T., Li, L., Liu, Y., Zhao, J., Wang, Y.: Deepgauge: Multi-granularity testing criteria for deep learning systems. In: ASE. pp. 120–131 (2018)

39. Metzen, J.H., Kumar, M.C., Brox, T., Fischer, V.: Universal adversarial perturbations against semantic image segmentation. In: ICCV. pp. 2774–2783 (2017)
40. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: CVPR. pp. 86–94 (2017)
41. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: A simple and accurate method to fool deep neural networks. In: CVPR. pp. 2574–2582 (2016)
42. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav trackings. In: ECCV. pp. 445–461 (2016)
43. Mueller, M., Bibi, A., Giancola, S., Al-Subaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: ECCV. pp. 310–327 (2018)
44. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: CVPR. pp. 4293–4302 (2016)
45. Papernot, N., McDaniel, P.D., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. IEEE European Symposium on Security and Privacy (EuroS P) pp. 372–387 (2016)
46. Qin, Y., Carlini, N., Goodfellow, I., Cottrell, G., Raffel, C.: Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In: arXiv:1903.10346 (2019)
47. Ren, S., Deng, Y., He, K., Che, W.: Generating natural language adversarial examples through probability weighted word saliency. In: ACL. pp. 1085–1097 (2019)
48. Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., Shen, C., Lau, R.W., Yang, M.H.: Vital:visual tracking via adversarial learning. In: CVPR. pp. 8990–8999 (2018)
49. Sun, J., Zhang, T., Xie, X., Ma, L., Zheng, Y., Chen, K., Liu, Y.: Stealthy and efficient adversarial attacks against deep reinforcement learning. In: AAAI. pp. 5883–5891 (2020)
50. Sun, Y., Sun, C., Wang, D., Lu, H., He, Y.: Roi pooled correlation filters for visual tracking. In: CVPR. pp. 5776–5784 (2019)
51. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: arXiv:1312.6199 (2013)
52. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: CVPR. pp. 1328–1338 (2019)
53. Wang, X., Li, C., Luo, B., Tang, J.: Sint++:robust visual tracking via adversarial positive instance generation. In: CVPR. pp. 4864–4873 (2018)
54. Wei, X., Liang, S., Chen, N., Cao, X.: Transferable adversarial attacks for image and video object detection. In: IJCAI. pp. 954–960 (2019)
55. Wei, X., Zhu, J., Yuan, S., Su, H.: Sparse adversarial perturbations for videos. In: AAAI. pp. 8973–8980 (2019)
56. Wiyatno, R.R., Xu, A.: Physical adversarial textures that fool visual object tracking. In: arXiv:1904.11042 (2019)
57. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE TPAMI **37**(9), 1834–1848 (2015)
58. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.L.: Adversarial examples for semantic segmentation and object detection. In: ICCV. pp. 1378–1387 (2017)
59. Xie, X., Ma, L., Juefei-Xu, F., Xue, M., Chen, H., Liu, Y., Zhao, J., Li, B., Yin, J., See, S.: Deephunter: A coverage-guided fuzz testing framework for deep neural networks. In: ISSTA. pp. 146–157 (2019)
60. Zhang, H., Zhou, H., Miao, N., Li, L.: Generating fluent adversarial examples for natural languages. In: ACL. pp. 5564–5569 (2019)
61. Zhang, P., Guo, Q., Feng, W.: Fast and object-adaptive spatial regularization for correlation filters based tracking. Neurocomputing **337**, 129 – 143 (2019)
62. Zhang, Y., Wang, L., Qi, J., Wang, D., Feng, M., Lu, H.: Structured siamese network for real-time visual tracking. In: ECCV. pp. 355–370 (2018)

63. Zhang, Z., Peng, H.: Deeper and wider siamese networks for real-time visual tracking. In: CVPR. pp. 4586–4595 (2019)
64. Zhao, Y., Zhu, H., Liang, R., Shen, Q., Zhang, S., Chen, K.: Seeing isn't believing: Practical adversarial attack against object detectors. In: CCS. pp. 1989–2004 (2019)
65. Zhu, Z., Wang, Q., Li, B., Wei, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: ECCV. pp. 103–119 (2018)