Learning to Adversarially Blur Visual Object Tracking

Qing Guo[1,5*], Ziyi Cheng[2*], Felix Juefei-Xu[3], Lei Ma[4†], Xiaofei Xie[5†], Yang Liu[5,6], Jianjun Zhao[2]

[1] College of Intelligence and Computing, Tianjin University, China,
[2] Kyushu University, Japan, [3] Alibaba Group, USA, [4] University of Alberta, Canada,
[5] Nanyang Technological University, Singapore, [6] Zhejiang Sci-Tech University, China

Abstract

*Motion blur caused by the moving of the object or camera during the exposure can be a key challenge for visual object tracking, affecting tracking accuracy significantly. In this work, we explore the robustness of visual object trackers against motion blur from a new angle, i.e., adversarial blur attack (ABA). Our main objective is to online transfer input frames to their natural motion-blurred counterparts while misleading the state-of-the-art trackers during the tracking process. To this end, we first design the motion blur synthesizing method for visual tracking based on the generation principle of motion blur, considering the motion information and the light accumulation process. With this synthetic method, we propose optimization-based ABA (OP-ABA) by iteratively optimizing an adversarial objective function against the tracking w.r.t. the motion and light accumulation parameters. The OP-ABA is able to produce natural adversarial examples but the iteration can cause heavy time cost, making it unsuitable for attacking real-time trackers. To alleviate this issue, we further propose one-step ABA (OS-ABA) where we design and train a joint adversarial motion and accumulation predictive network (JAMANet) with the guidance of OP-ABA, which is able to efficiently estimate the adversarial motion and accumulation parameters in a one-step way. The experiments on four popular datasets (*e.g.*, OTB100, VOT2018, UAV123, and LaSOT) demonstrate that our methods are able to cause significant accuracy drops on four state-of-the-art trackers with high transferability. Please find the source code at https://github.com/tsingqguo/ABA*

1. Introduction

Visual object tracking (VOT) has played an integral part in multifarious computer vision applications nowadays ranging from augmented reality [1, 46] to video surveillance [47], from human-computer interaction [36, 32] to traffic control [49], *etc*. Since the infusion of deep learning, VOT has become more powerful in terms of both algorithmic performance and efficiency [20], leading to the more pervasive deployment of VOT-enabled on-device applications. However, the VOT can still exhibit robustness brittleness when faced with less ideal video feed. Among many known degrading factors such as illumination variations, noise variations, *etc*., motion blur is perhaps one of the most important adverse factors for visual object tracking, which is caused by the moving of the object or camera during the exposure, and can severely jeopardize tracking accuracy [18]. Most of the existing benchmarks [28, 50, 35] only indicate whether a video or a frame contains motion blur or not and this piece of information is still insufficient to analyze the influence from motion blur by means of controlling all the variables, *e.g*., eliminating other possible interference from other degradation modes, which may lead to incomplete conclusions regarding the effects of motion blur in these benchmarks.

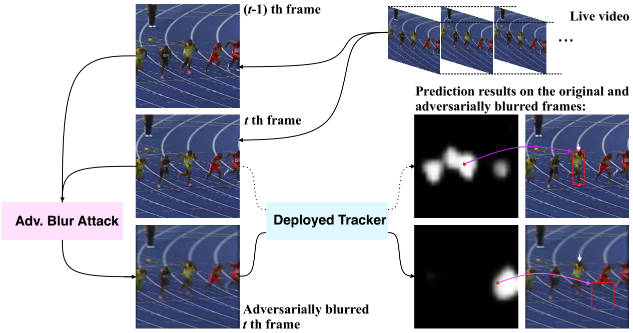
(*t*-1) th frame
Live video
...
Prediction results on the original and adversarially blurred frames:
t th frame
Adv. Blur Attack
Deployed Tracker
Adversarially blurred *t* th frame

Figure 1: An example of our adversarial blur attack against a deployed tracker, *e.g*., SiamRPN++ [30]. Two adjacent frames are fed to our attack and it generates an adversarially blurred frame that misleads the tracker to output an inaccurate response map.

Moreover, the currently limited datasets, albeit being large-scale, cannot well cover the diversity of motion blur in the real world because motion blur is caused by camera and object moving in the scene which is both dynamic and unknown. Existing motion blur generation methods cannot

[*] Qing Guo and Ziyi Cheng are co-first authors and contribute equally.
[†] Lei Ma and Xiaofei Xie are corresponding authors (ma.lei@acm.org, xfxie@ntu.edu.sg).

thoroughly reveal the malicious or unintentional threat to visual object tracking, *i.e.*, they can only produce natural motion blur which falls short of exposing the adversarial brittleness of the visual object tracker. As a result, it is necessary to explore a novel motion blur synthetic method for analyzing the robustness of the visual object trackers, which should not only generate natural motion-blurred frames but also embed maliciously adversarial or unintentional threats.

In this work, we investigate the robustness of visual trackers against motion blur from a new angle, that is, adversarial blur attack (ABA). Our main objective is to online transfer input frames to their natural motion-blurred counterparts while misleading the state-of-the-art trackers during the tracking process. We show an intuitive example in Fig. 1. To this end, we first design the motion blur synthesizing method for visual tracking based on the generation principle of motion blur, considering the motion information and the light accumulation process. With this synthetic method, we further propose *optimization-based ABA (OP-ABA)* by iteratively optimizing an adversarial objective function against the tracking w.r.t. the motion and light accumulation parameters.

The OP-ABA is able to produce natural adversarial examples but the iteration can lead to a heavy time-consuming process that is not suitable for attacking the real-time tracker. To alleviate this issue, we further propose *one-step ABA (OS-ABA)* where we design and train a *joint adversarial motion and accumulation predictive network (JAMANet)* with the guidance of OP-ABA, which is able to efficiently estimate the adversarial motion and accumulation parameters in a one-step way. The experiments on four popular datasets (*e.g.*, OTB100, VOT2018, UAV123, and LaSOT) demonstrate that our methods are able to cause significant accuracy drops on four state-of-the-art trackers while keeping the high transferability. To the best of our knowledge, this is the very first attempt to study the adversarial robustness of VOT and the findings will facilitate future-generation visual object trackers to perform more robustly in the wild.

2. Related Work

Visual object tracking (VOT). VOT is an important task in computer vision. Recently, a great number of trackers, which extract features with convolutional neural networks (CNNs), are proposed and achieve amazing performance.

Among these works, Siamese network-based methods [2, 14, 31, 19, 55, 45, 54, 44] offline train Siamese networks and conduct online matching between search regions and the object template, which are significantly fast with high tracking performance. In particular, SiamRPN [31, 30] embed the regional proposal network [40] in the naive Siamese tracker [2], allowing high-efficient estimation of the object's aspect ratio variation and achieving state-of-the-art tracking accuracy. After that, some works use historical frames to online update tracking models. For example, DiMP [3] collects past frames' features and online predict convolution kernels that can estimate object's position. Furthermore, PrDiMP [9] improves the loss function with KL divergence and information entropy from the perspective of probability distribution. KYS [4] considers the correlation between previous frames and the current frame. These trackers run beyond real time and get top accuracy on several benchmarks. Although great progress has been achieved, there are few works studying their robustness to motion blur. In this work, we identify a new way to achieve this goal by actively synthesizing adversarially motion blur to fool the state-of-the-art trackers.

Motion blur synthesis. In VOT task, motion blur is a very common scene due to the high-speed movement of the target. It is usually used to evaluate the quality of the trackers [50, 13, 18]. In recent years, motion blur synthesis has been extensively studied in the rendering community [38, 18]. However, these methods usually require a complete understanding of the speed and depth of the scene as input. In order to get more realistic and high-quality images with motion blur, Brooks *et al.* [5] identify a simple solution that warps two instant images by optical flow [42, 26] and fuses these intermediate frames with specific weights, to synthesize a blur picture. This method is to synthesize realistic motion blur for the deblurring task while our work is used for adversarially blurring the frames for tracking. Another related work, *i.e.*, ABBA [21], takes a single image as its input and generates visually natural motion-blurred adversarial example to fool the deep neural network-based classification. Specifically, ABBA simulates the motion by adversarially shifting the object and background, respectively, neglecting the real motion in the scene. Different from ABBA, our approach focuses on visual object tracking with real object movement indicated by two adjacent frames. Recently, some techniques [4, 9, 44] have been proposed to counter the interference of the environment. To this end, our method is proposed to better evaluate the robustness of these VOTs.

Adversarial attack. Extensive works have proved that state-of-the-art deep neural networks are still vulnerable to adversarial attacks by adding visually imperceptible noises or natural degradation to original images [16, 43, 8, 15, 21]. FGSM [16] perturbs normal examples along the gradient direction via the fast gradient sign method. MI-FGSM [11] integrates momentum term into the iterative process that can help stabilize the update directions. C&W [6] introduces three new attacks for different norms (L_0, L_2, L_∞) through iterative optimization. However, the above methods are unable to meet the real-time requirements due to the limited speed [22]. To realize the efficient attacking, [52, 51] propose one-step attacks by offline training on the targeted model. However, these methods are designed for the classification task and could not attack trackers directly.

More recently, some works have been proposed to attack visual object tracking. PAT [48] generates physical adversar-

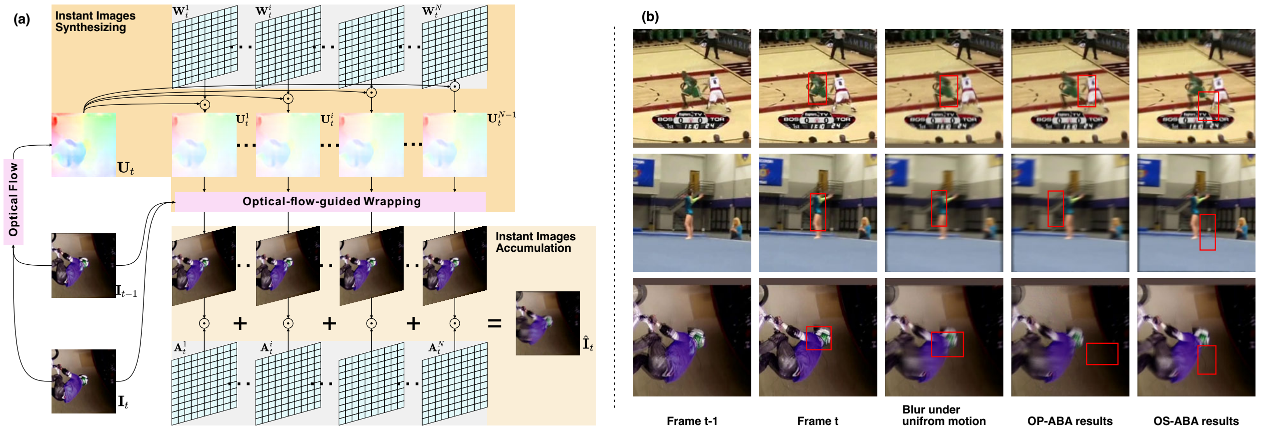

Figure 2: (a) shows the motion blur synthesizing process with two frames, *i.e.*, $\mathbf{I}_t$ and $\mathbf{I}_{t-1}$, and two sets of variables, *i.e.*, $\{\mathcal{A}_t^i\}$ and $\{\mathcal{W}_t^i\}$, should be determined for attacking. (b) shows three cases of the normal blur under uniform motion, the OP-ABA blurring results, and OS-ABA blurring results.

ial textures via a white-box attack. SPARK [22] studies how to adapt existing adversarial attacks on tracking. Chen *et al*. [7] propose to add adversarial perturbations on the template at the initial frame. CSA [52] raises a one-step method and makes objects invisible to trackers by forcing the predicted bounding box to shrink. Different from above works, we employ motion blur to perform adversarial attack. Our work is designed to address three challenges: how to synthesize natural motion blur that meets the motion of object and background in the video; how to make the blurred frame fool state-of-the-art trackers easily; how to perform the adversarial blur attack efficiently. To best of our knowledge, this is the very first attempt in the community of adversarial attack.

3. Adversarial Blur Attack against Tracking

In this section, we first study how to synthesize natural motion blur under the visual tracking task in Sec. 3.1 and summarize the variables that should be solved to perform attacks. Then, we propose the optimization-based ABA (OP-ABA) in Sec. 3.2 with a novel objective function to guide the motion blur generation via the iterative optimization process. To allow high-efficient attack for real-time trackers, we further propose one-step ABA (OS-ABA) in Sec. 3.3 by training a new designed *joint motion and kernel predictive network* under the supervision of the objective function of OP-ABA. Finally, we summarize the attacking details with OP-ABA and OS-ABA in Sec. 3.4.

3.1. Motion Blur Synthesizing for Visual Tracking

In a typical tracking process, given the t-th frame $\mathbf{I}_t$ of a live video and an object template specified at the first frame, a tracker uses a pre-trained deep model $\phi_{\theta_t}(\mathbf{I}_t)$ to predict the location and size of the object (*i.e.*, the bounding box tightly warping the object) in this frame where θ_t denotes the template-related parameter and can be updated during the tracking process. For the adversarial blur attack, we aim to generate a motion-blurred counterpart of $\mathbf{I}_t$, which is able to fool the tracker to estimate the incorrect bounding box of the object while having the natural motion-blur pattern.

To this end, we review the generation principle of realistic motion blur [37, 39, 5, 21, 18]: the camera sensor captures an image by receiving and accumulating light during the shutter procedure. The light at each time can be represented as an instant image, and there are a series of instant images for the shutter process. When the objects or background move, the light accumulation will cause blurry effects, which can be approximated by averaging the instant images.

Under the above principle, when we want to adversarially blur $\mathbf{I}_t$, we need to do two things: *First*, synthesizing the instant images during the shutter process and letting them follow the motion of object and background in the video; *Second*, accumulating all instant images to get the motion-blurred $\mathbf{I}_t$. The main challenge is how to make the two steps adversarially tunable to fool the tracker easily while preserving the natural motion blur pattern.

For the first step, we propose to generate the instant images under the guidance of the optical flow $\mathbf{U}_t$ that describes the pixel-wise moving distance and direction between $\mathbf{I}_t$ and its neighbor $\mathbf{I}_{t-1}$. Specifically, given two neighboring frames in a video, *e.g.*, $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$, we regard them as the start and end time stamps for camera shutter process, respectively. Assuming there are N instant images, we denote them as $\{\mathbf{I}_t^i\}_{i=1}^N$ where $\mathbf{I}_t^1 = \mathbf{I}_{t-1}$ and $\mathbf{I}_t^N = \mathbf{I}_t$. Then, we calculate the optical flow $\mathbf{U}_t$ between $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$ and split it into $N-1$ sub-motions, *i.e.*, $\{\mathbf{U}_t^i\}_{i=1}^{N-1}$ where $\mathbf{U}_t^i$ represents the optical flow between $\mathbf{I}_t^i$ and $\mathbf{I}_t^{i+1}$. We define $\mathbf{U}_t^i$ as a scaled $\mathbf{U}_t$ with pixel-wise ratios (*i.e.*, $\mathbf{W}_t^i$)

$$\mathbf{U}_t^i = \mathbf{W}_t^i \odot \mathbf{U}_t, \tag{1}$$

where $\mathbf{W}_t^i$ has the same size with $\mathbf{U}_t$ and $\odot$ denotes the pixel-wise multiplication. All elements in $\mathbf{W}_t^i$ range from zero to one and we constraint the summation of $\{\mathbf{W}_t^i\}_{i=1}^N$ at the same position to be one, *i.e.*, $\forall \mathbf{p}, \sum_i^{N-1} \mathbf{W}_t^i[\mathbf{p}] = 1$

where $\mathbf{W}_t^i[\mathbf{p}]$ denotes the $\mathbf{p}$-th element in $\mathbf{W}_t^i$. Note that, the ratio matrices, *i.e.*, $\{\mathbf{W}_t^i\}_{i=1}^N$, determine the motion pattern. For example, if we have $\forall \mathbf{p}, \{\mathbf{W}_t^i[\mathbf{p}] = \frac{1}{N-1}\}_{i=1}^{N-1}$ and can calculate the sub-motions by $\{\mathbf{U}_t^i = \frac{1}{N-1}\mathbf{U}_t\}_{i=1}^{N-1}$, all pixels follow the uniform motion.

With Eq. (1), we get all sub-motions (*i.e.*, $\{\mathbf{U}_t^i\}_{i=1}^{N-1}$) and produce the instant images by warping $\mathbf{I}_t$ w.r.t. different optical flows. For example, we synthesize $\mathbf{I}_t^i$ by

$$\mathbf{I}_t^i = \frac{1}{2}\text{warp}(\mathbf{I}_{t-1}, \sum_{j=1}^{i-1} \mathbf{W}_t^j \odot \mathbf{U}_t^j) + \frac{1}{2}\text{warp}(\mathbf{I}_t, \sum_{j=i}^{N-1} \mathbf{W}_t^j \odot \mathbf{U}_t^j), \tag{2}$$

where $\sum_{j=1}^{i-1} \mathbf{W}_t^j \odot \mathbf{U}_t^j$ represents the optical flow between $\mathbf{I}_{t-1}$ and $\mathbf{I}_t^i$ while $\sum_{j=i}^{N-1} \mathbf{W}_t^j \odot \mathbf{U}_t^j$ denotes the optical flow between $\mathbf{I}_t^i$ and $\mathbf{I}_t$ The function warp$(\cdot)$ is to wrap the $\mathbf{I}_{t-1}$ or $\mathbf{I}_t$ according to the corresponding optical flow, and uses the implementation in [21] for spatial transformer network.

For the second step, after getting $\{\mathbf{I}_t^i\}_{i=1}^N$, we can synthesize the motion-blurred $\mathbf{I}_t$ by summing up the N instant images with pixel-wise accumulation weights $\{\mathbf{A}_i\}_{i=1}^N$

$$\hat{\mathbf{I}}_t = \sum_{i=1}^{N} \mathbf{A}_t^i \odot \mathbf{I}_t^i. \tag{3}$$

where $\mathbf{A}_t^i$ has the same size with $\mathbf{I}_t^i$ and all elements range from zero to one. For simulating realistic motion blur, all elements of $\mathbf{A}_t^i$ are usually fixed as $\frac{1}{N}$, which denotes the accumulation of all instant images.

Overall, we represent the whole blurring process via Eqs. (3) and (2) as $\hat{\mathbf{I}}_t = \text{Blur}(\mathbf{I}_t, \mathbf{I}_{t-1}, \mathcal{W}_t, \mathcal{A}_t)$. To perform adversarial blur attack for the frame $\mathbf{I}_t$, we need to solve two sets of variables, *i.e.*, $\mathcal{W}_t = \{\mathbf{W}_t^i\}_{i=1}^{N-1}$ determining the motion pattern and $\mathcal{A}_t = \{\mathbf{A}_t^i\}_{i=1}^N$ deciding the accumulation strategy. In Sec. 3.2, we follow the existing adversarial attack pipeline and propose the optimization-based ABA by defining and optimizing a tracking-related objective function to get $\mathcal{W}_t$ and $\mathcal{A}_t$. In Sec. 3.3, we design a network to predict $\mathcal{W}_t$ and $\mathcal{A}_t$ in a one-step way.

3.2. Optimization-based Adversarial Blur Attack

In this section, we propose to solve $\mathcal{W}_t$ and $\mathcal{A}_t$ by optimizing the tracking-related objective function. Specifically, given the original frame $\mathbf{I}_t$, a tracker can estimate a response or classification map by $\mathbf{Y}_t = \phi_{\theta_t}(\mathbf{I}_t)$ whose maximum indicating the object's position in the $\mathbf{I}_t$. Our attack aims to generate a blurred $\mathbf{I}_t$ (*i.e.*, $\hat{\mathbf{I}}_t = \text{Blur}(\mathbf{I}_t, \mathbf{I}_{t-1}, \mathcal{W}_t, \mathcal{A}_t)$) to let the predicted object position indicated by $\hat{\mathbf{Y}}_t = \phi_{\theta_t}(\hat{\mathbf{I}}_t)$ be far away from the original one indicated by $\mathbf{Y}_t$.

To this end, we optimize $\mathcal{W}_t$ and $\mathcal{A}_t$ by minimizing

$$\arg\min_{\mathcal{W}_t, \mathcal{A}_t} J(\phi_{\theta_t}(\text{Blur}(\mathbf{I}_t, \mathbf{I}_{t-1}, \mathcal{W}_t, \mathcal{A}_t)), \mathbf{Y}_t^*)$$
$$\text{subject to } \forall \mathbf{p}, \forall i, \sum_{i}^{N-1} \mathbf{W}_t^i[\mathbf{p}] = 1, \sum_{i}^{N} \mathbf{A}_t^i[\mathbf{p}] = 1, \tag{4}$$

where the two constraints on $\mathbf{W}_t^i$ and $\mathbf{A}_t^i$ make sure the synthetic motion blur does not have obvious distortions. The function $J(\cdot)$ is a distance function and is set as L_2. The regression target $\mathbf{Y}_t^*$ denotes the desired response map and is obtained under the guidance of the original $\mathbf{Y}_t$. Specifically, with the original response map $\mathbf{Y}_t$, we know the object's position and split $\mathbf{Y}_t$ into two regions the object region and background region according to the object size. Then, we can find the position (*e.g.*, $\mathbf{q}$) having the highest response score at the background region of $\mathbf{Y}_t$ and then we set $\mathbf{Y}^*[\mathbf{q}] = 1$ and other elements of $\mathbf{Y}^*$ to be zero. Note that, above setup is suitable for regression-based trackers, *e.g.*, DiMP and KYS, and can be further adapted to attack classification-based trackers, *e.g.*, SiamRPN++, by setting $J(\cdot)$ as the cross-entropy loss function and $\mathbf{Y}^*[\mathbf{q}] = 1$ with its other elements to be -1.

Following the common adversarial attacks [17, 12, 22, 21], we can solve Eq. (4) via the signed gradient descent and update the $\mathcal{W}_t$ and $\mathcal{A}_t$ iteratively with specified step size and iterative number. We show the synthesized motion blur of OP-ABA in Fig. 2. Clearly, OP-ABA is able to synthesize natural motion-blurred frames that have a similar appearance to the normal motion blur.

3.3. One-Step Adversarial Blur Attack

To allow efficient adversarial blur attack, we propose to predict the motion and accumulation weights (*i.e.*, $\mathcal{W}_t$ and $\mathcal{A}_t$) with a newly designed network denoted as *joint adversarial motion and accumulation predictive network (JAMANet)* in a one-step way, which is pre-trained through the objective function Eq. (4) and a naturalness-aware loss function. Specifically, we use JAMANet to process the neighboring frames (*i.e.*, $\mathbf{I}_t$ and $\mathbf{I}_{t-1}$) and predict the $\mathcal{W}_t$ and $\mathcal{A}_t$, respectively. Meanwhile, we also employ a pre-trained network to estimate the optical flow $\mathbf{U}_t$ between $\mathbf{I}_t$ and $\mathbf{I}_{t-1}$. Here, we use the PWCNet [42] since it achieves good results on diverse scenes. Then, with Eq. (2)-(3), we can obtain the motion-blurred frame $\hat{\mathbf{I}}_t$. After that, we feed $\hat{\mathbf{I}}_t$ into the loss functions and calculate gradients of parameters of JAMANet to perform optimization. We show the framework in Fig. 3.

Architecture of JAMANet. We first build two parameter sets with constant values, which are denoted as $\mathcal{A}_{\text{norm}} = \{\mathbf{A}_{\text{norm}}^i\}$ and $\mathcal{W}_{\text{norm}} = \{\mathbf{W}_{\text{norm}}^i\}$. All elements in $\mathcal{A}_{\text{norm}}$ and $\mathcal{W}_{\text{norm}}$ are fixed as $\frac{1}{N}$ and $\frac{1}{N-1}$, respectively. We then use $\mathcal{W}_{\text{norm}}$, $\mathbf{I}_{t-1}$, and $\mathbf{I}_t$ to generate N instant images through Eq. (2). JAMANet is built based on the U-Net architecture [41] but contains two decoder branches, which

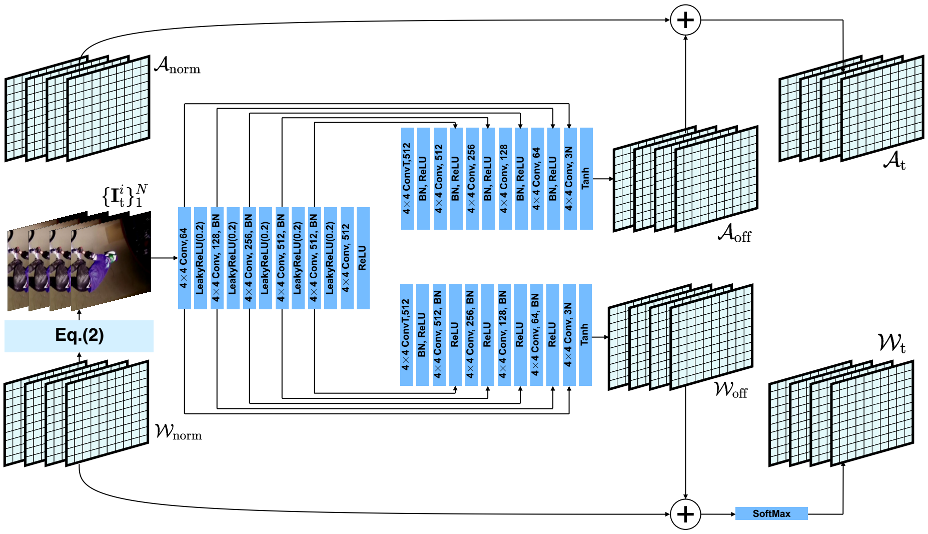

Figure 3: Architecture of JAMANet.

is fed the N instant images $\{\mathbf{I}_t^i\}_{i=1}^N$ and outputs the offsets w.r.t. $\mathcal{W}_{\text{norm}}$ and $\mathcal{A}_{\text{norm}}$. We name them as $\mathcal{W}_{\text{off}}$ and $\mathcal{A}_{\text{off}}$. The input $\{\mathbf{I}_t^i\}_{i=1}^N$ is size of $(N, 3, H, W)$. We resize it to $(1, 3N, H, W)$ and normalize the values to the range of -1 to 1. The architecture is a full convolutional encoder/decoder model with skip connections. In encoder stage, we use six convolutions with the kernel size 4x4 and the LeakyReLU [33] activation function. Unlike the standard U-Net, JAMANet has two decoders. Specifically, one branch is set to estimate the $\mathcal{A}_{\text{off}}$, containing six transposed convolutions [53] with the latest activation function as Tanh. We can calculate the final $\mathcal{A}_t$ through $\mathcal{A}_t = \mathcal{A}_{\text{norm}} + \mathcal{A}_{\text{off}}$. Another branch is to predict $\mathcal{W}_{\text{off}}$ and get $\mathcal{W}_t = \mathcal{W}_{\text{norm}} + \mathcal{W}_{\text{off}}$. This architecture is the same with the previous one but following an Softmax for catering to constraints of Eq. (4)[1].

Loss functions. We train the JAMANet with two loss functions:

$$\mathcal{L} = \mathcal{L}_{\text{adv}} + \lambda \mathcal{L}_{\text{natural}}, \tag{5}$$

where the first loss function, *i.e.*, $\mathcal{L}_{\text{adv1}}$, is set to the objective function in Eq. (4) to make sure the background content instead of the object be highlighted. Note that, this loss function means to enhance the capability of adversarial attack, that is, misleading the raw trackers. It, however, neglects the naturalness of adversarial blur. To this end, we set the loss function $\mathcal{L}_{\text{natural}}$ as

$$\mathcal{L}_{\text{natural}} = \sum_i^N \|\mathbf{A}_t^i - \mathbf{A}_{\text{norm}}^i\|_2. \tag{6}$$

This loss function encourages the estimated accumulation parameters to be similar to normal ones, leading to natural motion blur.

Training details. We use GOT-10K [25] as our training dataset, which includes 10,000+ sequences and 500+ object classes. For each video in GOT-10K [25], we set the first frame as template and take two adjacent frames as an image pair, *i.e.*, $(\mathbf{I}_{t-1}, \mathbf{I}_t)$. We select eight image pairs from each video. The template and two adjacent frames make up a training sample. Here, we implement the OS-ABA for attacking two trackers, *i.e.*, SiamRPN++ [30] with ResNet50 and MobileNetv2, respectively. In the experiment, we show that OS-ABA has strong transferability against other state-of-the-art trackers. During the training iteration, we first calculate the template's embedding to construct tracking model ϕ_{θ_t} and the original response map $\mathbf{Y}_t$ (*i.e.*, the positive activation map of SiamRPN++). Then, we get $\mathbf{Y}_t^*$ and initialize the blurred frame via $\text{Blur}(\mathbf{I}_t, \hat{\mathbf{I}}_{t-1})$. We can calculate the loss via Eq. (5) and obtain the gradients of the JAMANet via backpropagation for parameter updating. We train the JAMANet for 10 epochs, requiring a total of about 9 hours on a single Nvidia RTX 2080Ti GPU. We use the Adam [27] with the learning rate of 0.0002 to optimize network parameters, and the loss weight λ equals 0.001.

3.4. Attacking Details

Intuitively, given a targeted track, we can attack it by blurring each frame through OP-ABA and OS-ABA during the online tracking process, as shown in Fig. 1. The attack could be white-box, that is, the tracking model in Eq. (4) is the same as the targeted track, leading to high accuracy drop. It also could be black-box also known as the transferability, that is, the tracking model Eq. (4) is different from the targeted one. Note that, OP-ABA is based on iterative optimization and is time-consuming, thus we conduct OP-ABA every five frames while performing OS-ABA for all frames. In practice, we perform the blurring on the search regions

[1] To let $\mathcal{A}_t$ also meet the constraints, for any pixel $\mathbf{p}$, we first select the element $j = \arg\min_{i,i\in[1,N]} \mathbf{A}_{\text{off}}^i[\mathbf{p}]$ and then set $\mathbf{A}_t^j[\mathbf{p}] = 1 - \sum_{i,i\neq j}^N \mathbf{A}_t^i[\mathbf{p}]$.

Table 1: Attacking results of OP-ABA and OS-ABA against SiamRPN++ with ResNet50 and MobileNetv2 on OTB100 and VOT2018. The best results are highlighted by **red** color.

SiamRPN++	Attacks	OTB100				VOT2018	
		Org. Prec.	Prec. Drop ↑	Org. Succ.	Succ. Drop ↑	Org. EAO	EAO Drop ↑
ResNet50	OP-ABA	87.8	**41.7**	66.5	**31.2**	0.415	**0.375**
	OS-ABA	87.8	32.5	66.5	28.1	0.415	0.350
MobNetv2	OP-ABA	86.4	**49.6**	65.8	**37.6**	0.410	**0.384**
	OS-ABA	86.4	37.3	65.8	30.1	0.410	0.338

Table 2: Attacking results of OP-ABA and OS-ABA against SiamRPN++ with ResNet50 and MobileNetv2 on UAV123 and LaSOT. The best results are highlighted by **red** color.

SiamRPN++	Attacks	UAV123				LaSOT			
		Org. Prec.	Prec. Drop. ↑	Org. Succ.	Succ. Drop ↑	Org. Prec.	Prec. Drop ↑	Org. Succ.	Succ. Drop ↑
ResNet50	OP-ABA	80.4	**30.4**	61.1	**23.1**	49.0	**28.7**	49.7	25.2
	OS-ABA	80.4	29.6	61.1	19.9	49.0	26.8	49.7	**26.4**
MobNetv2	OP-ABA	80.2	**34.7**	60.2	**26.9**	44.6	**29.7**	44.7	**28.1**
	OS-ABA	80.2	31.9	60.2	24.0	44.6	22.5	44.7	18.7

between two frames to accelerate the attacking speed. Specifically, at the frame t, we crop a search region centered at the detected object as the $\mathbf{I}_t$. At the same time, we crop a region from the previous frame at the same position as the $\mathbf{I}_{t-1}$. Then, we use the PWCNet [42] to calculate optical flow. We get the original response map with the targeted tracker and $\mathbf{I}_t$ if we employ the OP-ABA as the attack method. After that, we can conduct the OP-ABA or OS-ABA to generate the adversarial blurred frame. In terms of the OP-ABA, we set the iteration number to be 10 and the step sizes for updating $\mathcal{W}_t$ and $\mathcal{A}_t$ are set as 0.002 and 0.0002, respectively. The number of intermediate frames N is fixed as 17 for both OP-ABA and OS-ABA.

4. Experimental Results

We design experiments to investigate three aspects: *First*, we validate the effectiveness of our two methods against state-of-the-art trackers on four public tracking benchmarks in Sec. 4.2. *Second*, we design ablation experiments to validate the influences of $\mathcal{A}_t$ and $\mathcal{W}_t$ in Sec. 4.3. *Third*, we compare our method with state-of-the-art tracking attacks about their transferability and frame quality in Sec. 4.4.

4.1. Setups

Datasets. We evaluated adversarial blur attack on four popular datasets, *i.e.*, VOT2018 [28], OTB100 [50], UAV123 [35], and LaSOT [13]. VOT2018 and OTB100 are widely used datasets containing 100 videos and 60 videos, respectively. LaSOT is a recent large-scale tracking benchmark, which contains 280 videos. UAV123 [35] focuses on tracking the object captured by unmanned aerial vehicle's camera, including 123 videos.

Tracking models. We conduct attack against state-of-the-art trackers including SiamRPN++ [30] with ResNet50 [23] and MobileNetv2 [24], DiMP [3] with ResNet50 and ResNet18, and KYS [4]. Specifically, we validate the white-box attack with OP-ABA and OS-ABA against SiamRPN++ [30] with ResNet50 [23] and MobileNetv2 in Sec. 4.2 where the motion-blurred frames are guided by the targted tracker's model itself. We choose SiamRPN++ [30] since it is a classic tracker for Siamese network-based methods [44, 31, 10, 2, 19] which achieves excellent tracking accuracy and real-time tracking speed. We also conduct transferability experiments by using the motion blur crafted from SiamRPN++ with ResNet50 to attack other trackers.

Metrics. In terms of the OTB100, UAV123 and LaSOT datasets, we follow their common setups and use one pass evaluation (OPE) that contains two metrics *success rate* and *precision*. The former one is based on the intersection over union (IoU) between the ground truth bounding box and predicted one for all frames while the latter is based on the center location error (CLE) between the ground truth and prediction. Please refer to [50] for details. To evaluate the capability of attacking, we use the drop of success rate and precision for different attacks, which are denoted as Succ. Drop and Prec. Drop. The higher drops mean more effective attacking. In terms of VOT2018, it restarts trackers when the object is lost. Expected average overlap (EAO) [29] is the main criterion, evaluating both accuracy and robustness. Similar to Succ. Drop, we use the drop of EAO (*i.e.*, EAO Drop) for evaluating attacks. When comparing with other additive noise-based attacks, we use the BRISQUE [34] as the image quality assessment. An attack is desired to produce adversarial examples that are not only natural but also able to fool trackers. BRISQUE is a common metric to evaluate the naturalness of images and a smaller BRISQUE means a more natural image.

Baselines. There are several tracking attacks including cooling-shrinking attack (CSA) [52], SPARK [22], One-shot-based attack [7], and PAT [48]. Among them, CSA and SPARK have released their code. We select CSA and SPARK as the baselines.

Table 3: Speed and time cost of three attacks and SiamRPN++ with the ResNet50 and MobileNetv2.

SiamRPN++	Attackers	Org. FPS	Attack time (ms) ↓ per frame	Attack FPS ↑
ResNet50	OP-ABA OS-ABA	70.25 70.25	661.90 **42.97**	6.79 **17.62**
MobNetv2	OP-ABA OS-ABA	107.62 107.62	508.30 **40.88**	8.79 **19.96**

4.2. Validation Results

Attacking results. We attack two SiamRPN++ trackers that uses ResNet50 and MobileNetv2 as the backbone, respectively. The attacks results on the four public datasets are presented in Table 1 and 2, respectively. We observe that: ❶ Both OP-ABA and OS-ABA reduce the success rate and precision of the two targeted trackers significantly on all benchmarks. Specifically, on the OTB100 dataset, OP-ABA makes the precision and success rate of SiamRPN++ with ResNet50 reduce 41.7 and 31.2, respectively, almost fifty percent of the original scores. These results demonstrate that the proposed attacks are able to fool the state-of-the-art trackers effectively. ❷ Compared with OS-ABA, OP-ABA achieves higher precision drop since it targeted attack to a certain position during each optimization while OS-ABA generates a general blurred image to make objects invisible for trackers. In general, all the results indicate the effectiveness of OP-ABA and OS-ABA in misleading the tracking models by adversarial blur attack. ❸ Comparing the performance drop of SiamRPN++ (ResNet50) with SiamRPN++ (MobileNetv2), we observe that the former usually has relatively smaller precision or success rate drop under the same attack, hinting that the lighter model is fooled more easily. ❹ According to the visualization results shown in Fig. 4, we see that both methods are able to generate visually nature blurred frames that mislead the SiamRPN++. In general, OP-ABA contains some artifacts but is able to mislead the tracker more effectively than OS-ABA. In contrast, OS-ABA always generates more realistic motion blur than OP-ABA in all three cases.

Speed analysis. We test the time cost of OP-ABA and OS-ABA on the OTB100 and report the FPS of the SiamRPN++ trackers before and after attacking. As presented in Table 3 , we observe that OP-ABA would slow down the tracking speed significantly. For example, OP-ABA reduces the speed of SiamRPN++ with ResNet-50 from 63 FPS to 6.79 PFS due to the online optimization. Thanks to the one-step optimization via JAMANet in Sec. 3.3 , OS-ABA is almost ten times faster than OP-ABA according to the average attack time per frame. In consequence, OS-ABA achieved near real-time speed, *e.g.*, 17.62 FPS and 20.00 FPS, in attacking SiamRPN++ (ResNet50) and SiamRPN++ (MobileNetv2). In terms of the FPS after attacking, OS-ABA also about 3 times faster than OP-ABA.

Table 4: Effects of $\mathcal{W}_t$ and $\mathcal{A}_t$ to OP-ABA and OS-ABA by attacking SiamRPN++ (ResNet50) on OTB100. The best results are highlighted by **red** color.

Attackers	Succ. Rate	Succ. Drop ↑	Prec.	Prec. Drop ↑
Original Norm-Blur	66.5 65.3	0.0 1.2	87.8 86.2	0.0 1.6
OP-ABA w/o $\mathcal{A}_t$ OP-ABA w/o $\mathcal{W}_t$ OP-ABA	51.5 40.9 35,3	15.0 25.6 **31.2**	67.6 53.4 46.1	20.2 34.4 **41.7**
OS-ABA w/o $\mathcal{A}_t$ OS-ABA w/o $\mathcal{W}_t$ OS-ABA	61.0 41.6 38.4	5.5 24.9 **28.1**	80.8 58.3 55.3	7.0 29.5 **32.5**

4.3. Ablation Study

In this section, we discuss the influence of $\mathcal{W}_t$ and $\mathcal{A}_t$ to OP-ABA and OS-ABA by constructing two variants of them to attack SiamRPN++ (ResNet50) tracker on OTB100 dataset. Specifically, for both attacks, we only tune $\mathcal{A}_t$ and fix $\mathcal{W}_t$ as $\mathcal{W}_{\text{norm}}$, thus we get two variants OP-ABA w/o $\mathcal{W}_t$ and OS-ABA w/o $\mathcal{W}_t$. Similarly, we replace $\mathcal{A}_t$ with $\mathcal{A}_{\text{norm}}$ and adversarially tune $\mathcal{W}_t$, thus we get OP-ABA w/o $\mathcal{A}_t$ and OS-ABA w/o $\mathcal{A}_t$, respectively. Moreover, to demonstrate that it is the adversarial blur that reduces the performance, we build the 'Norm-Blur' attack. It synthesizes the motion blur with $\mathcal{A}_{\text{norm}}$ and $\mathcal{W}_{\text{norm}}$, representing the norm blur that may appear in the real world.

We summarize the results in Table 4 and Fig. 4 and have the following observes: ❶ When we fix the $\mathcal{W}_t$ or $\mathcal{A}_t$ for OP-ABA and OS-ABA, the success rate and precision drops decrease significantly, demonstrating that tuning both motion pattern (*i.e.*, $\mathcal{W}_t$) and accumulation strategy ($\mathcal{A}_t$) can benefit the adversarial blur attack. ❷ According to the variance of the performance drop, we see that tuning the accumulation strategy ($\mathcal{A}_t$) contributes more for effective attacks. For example, without tuning $\mathcal{A}_t$, the success rate drop reduces from 28.1 and 31.2 to 5.5 and 15.0 for OS-ABA and OP-ABA, respectively. ❸ SiamRPN++ are robust to the Norm-Blur with slight success rate and precision drops. In contrast, the adversarial blur causes a significant performance drop, demonstrating the adversarial blur does pose threat to visual object tracking. ❹ According to the visualization results in Fig. 4, we have similar conclusions with the quantitative results in Table 4: OP-ABA w/o $\mathcal{A}_t$ can generate motion-blurred frames but have little influence on the prediction accuracy. Once we tune $\mathcal{A}_t$, the tracker can be fooled effectively but some artifacts are also introduced.

4.4. Comparison with Other Attacks

In this section, we study the transferability of proposed attacks by comparing them with baseline attacks, *i.e.*, CSA [52] and SPARK [22]. Specifically, for all compared attacks, we use SiamRPN++(ResNet50) as the guidance to performance optimization or training. For example, we set ϕ_{θ_t} in the objective function of OP-ABA (*i.e.*, Eq. (4)) as the model

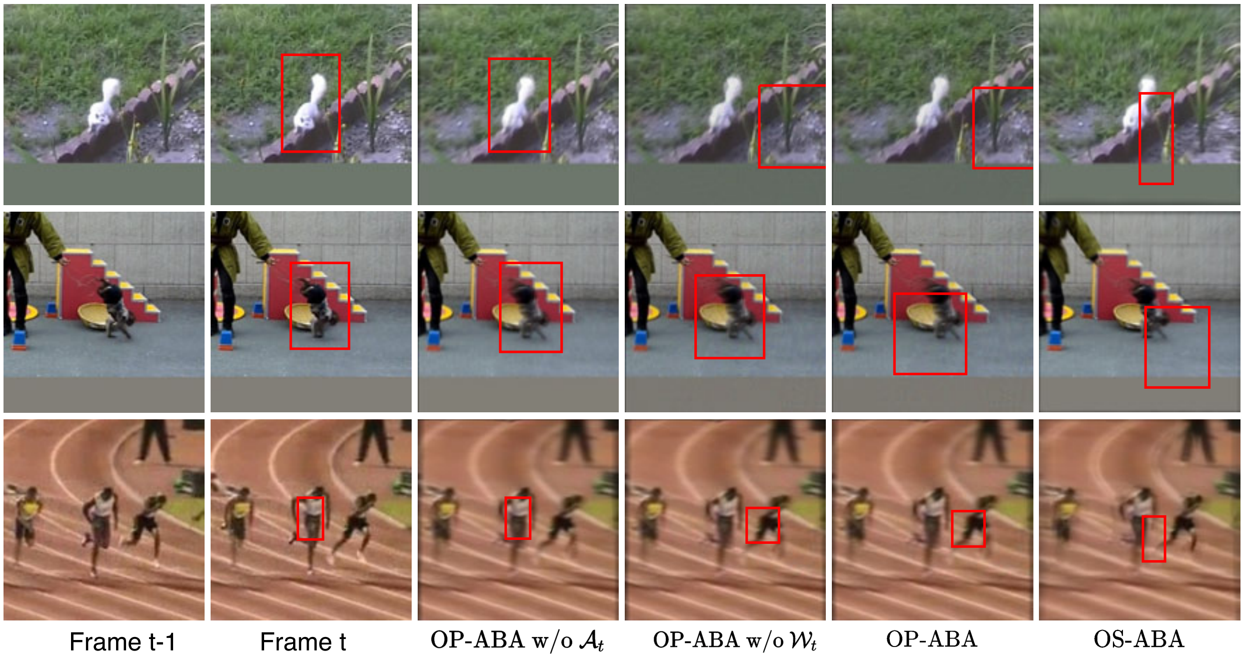

Figure 4: Three visualization results of OP-ABA w/o $\mathcal{A}_t$, OP-ABA w/o $\mathcal{W}_t$, OP-ABA, and OS-ABA against SiamRPN++ (ResNet50). The corresponding tracking results are showed with **red** bounding boxes.

Table 5: Comparison results on transferability. Specifically, we use the adversarial examples crafted from SiamRPN++ (ResNet50) to attack four state-of-the-art trackers including SiamRPN++ (MobileNetv2) [30], DiMP50 [3], DiMP18 [3], and KYS [4] on OTB100. We also calulate the average BRISQUE values of all adversarial examples.

Trackers	SiamRPN++ (MobNetv2)	DiMP50	DiMP18	KYS	BRISQUE ↓
Org. Prec.	86.4	89.2	87.1	89.5	20.15
CSA	0.2	3.4	2.7	0.8	33.63
SPARK	**0.9**	2.0	1.0	0.9	24.78
OP-ABA	**2.5**	**6.6**	**10.3**	**7.9**	**21.39**
OS-ABA	0.2	**10.7**	**11.2**	**12.3**	**22.94**

of SiamRPN++(ResNet50). We report the precision drop after attacking in Table 5 and the BRISQUE as the image quality assessment for generated adversarial frames.

As shown in Table 5, we observe: ❶ Our methods, *i.e.*, OP-ABA and OS-ABA, achieve the best and second-best transferability (*i.e.*, higher precision drop) against DiMP50, DiMP18 [3], and KYS [4], hinting that our methods are more practical for black-box attacking. ❷ According to BRISQUE results, the adversarially blurred frames have smaller values than other adversarial examples, hinting that our methods are able to generate more natural frames since motion blur is a common degradation in the real world.

5. Conclusion

In this work, we proposed a novel adversarial attack against visual object tracking, *i.e.*, adversarial blur attack (ABA), considering the effects of motion blur instead of the noise against the state-of-the-art trackers. We first identified the motion blur synthesizing process during tracking based on which we proposed the optimization-based ABA (OP-ABA). This method fools the trackers by iteratively optimizing a tracking-aware objective but causes heavy time cost. We further proposed the one-step ABA by training a novel designed network to predict blur parameters in a one-step way. The attacking results on four public datasets, the visualization results, and comparison results demonstrated the effectiveness and advantages of our methods. This work not only reveals the potential threat of motion blur against trackers but also could work as a new way to evaluate the motion-blur robustness of trackers in the future.

Acknowledgments: This work is supported in part by JSPS KAKENHI Grant No.JP20H04168, JP19K24348, JP19H04086, JP21H04877, JST-Mirai Program Grant No.JPMJMI20B8, Japan. Lei Ma is also supported by Canada CIFAR AI Program and Natural Sciences and Engineering Research Council of Canada. The work was also supported by the National Research Foundation, Singapore under its the AI Singapore Programme (AISG2-RP-2020-019), the National Research Foundation, Prime Ministers Office, Singapore under its National Cybersecurity R&D Program (No. NRF2018NCR-NCR005-0001), NRF Investigatorship NRF-NRFI06-2020-0001, the National Research Foundation through its National Satellite of Excellence in Trustworthy Software Systems (NSOE-TSS) project under the National Cybersecurity R&D (NCR) Grant (No. NRF2018NCR-NSOE003-0001). We gratefully acknowledge the support of NVIDIA AI Tech Center (NVAITC) to our research.

References

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 2001. 1

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 2, 6

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pages 6181–6190, 2019. 2, 6, 8

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *European Conference on Computer Vision*, pages 205–221. Springer, 2020. 2, 6, 8

[5] Tim Brooks and Jonathan T. Barron. Learning to synthesize motion blur. In *CVPR*, 2019. 2, 3

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 2

[7] Xuesong Chen, Xiyu Yan, Feng Zheng, Yong Jiang, Shu-Tao Xia, Yong Zhao, and Rongrong Ji. One-shot adversarial attacks on visual tracking with dual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10176–10185, 2020. 3, 6

[8] Yupeng Cheng, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Shang-Wei Lin, Weisi Lin, Wei Feng, and Yang Liu. Pasadena: Perceptually aware and stealthy adversarial denoise attack. *IEEE Transactions on MultiMedia*, 2021. 2

[9] M. Danelljan, L. Van Gool, and R. Timofte. Probabilistic regression for visual tracking. In *CVPR*, pages 7181–7190, 2020. 2

[10] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 6

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2

[12] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *CVPR*, pages 9185–9193, 2018. 4

[13] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, pages 5369–5378, 2019. 2, 6

[14] H. Fan and H. Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, pages 7944–7953, 2019. 2

[15] Ruijun Gao, Qing Guo, Felix Juefei-Xu, Hongkai Yu, Xuhong Ren, Wei Feng, and Song Wang. Making images undiscoverable from co-saliency detection. *arXiv preprint arXiv:2009.09258*, 2020. 2

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *arXiv:1412.6572*, 2014. 4

[18] Q. Guo, W. Feng, R. Gao, Y. Liu, and S. Wang. Exploring the effects of blur and deblurring to visual object tracking. *IEEE Transactions on Image Processing*, 30:1812–1824, 2021. 1, 2, 3

[19] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang. Learning dynamic Siamese network for visual object tracking. In *ICCV*, pages 1781–1789, 2017. 2, 6

[20] Qing Guo, Ruize Han, Wei Feng, Zhihao Chen, and Liang Wan. Selective spatial regularization by reinforcement learned decision making for object tracking. *IEEE Transactions on Image Processing*, 29:2999–3013, 2020. 1

[21] Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Jian Wang, Bing Yu, Wei Feng, and Yang Liu. Watch out! motion is blurring the vision of your deep neural networks. In *Advances in Neural Information Processing Systems 34*, 2020. 2, 3, 4

[22] Qing Guo, Xiaofei Xie, Felix Juefei-Xu, Lei Ma, Zhongguo Li, Wanli Xue, Wei Feng, and Yang Liu. Spark: Spatial-aware online incremental attack against visual tracking. In *ECCV*, 2020. 2, 3, 4, 6, 7

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 6

[25] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 5

[26] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 2

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[28] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka ˇCehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1, 6

[29] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking vot2015 challenge results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1–23, 2015. 6

[30] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4277–4286, 2019. 1, 2, 5, 6, 8

[31] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin 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. 2, 6

[32] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020. 1

[33] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013. 5

[34] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012. 6

[35] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. *ECCV*, pages 445–461, 2016. 1, 6

[36] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 122–132, 2020. 1

[37] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 257–265, 2017. 3

[38] Fernando Navarro, Francisco J Serón, and Diego Gutierrez. Motion blur rendering: State of the art. In *Computer Graphics Forum*, volume 30, pages 3–26. Wiley Online Library, 2011. 2

[39] M. Noroozi, P. Chandramouli, and P. Favaro. Motion deblurring in the wild. In *Pattern Recognition*, pages 65–77, 2017. 3

[40] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 2, 4, 6

[43] Binyu Tian, Qing Guo, Felix Juefei-Xu, Wen Le Chan, Yupeng Cheng, Xiaohong Li, Xiaofei Xie, and Shengchao Qin. Bias field poses a threat to dnn-based x-ray recognition. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021. 2

[44] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6588, 2020. 2, 6

[45] Xiao Wang, Chenglong Li, Bin Luo, and Jin Tang. Sint++:robust visual tracking via adversarial positive instance generation. pages 4864–4873, 2018. 2

[46] Jianing Wei, Genzhi Ye, Tyler Mullen, Matthias Grundmann, Adel Ahmadyan, and Tingbo Hou. Instant motion tracking and its applications to augmented reality. *arXiv preprint arXiv:1907.06796*, 2019. 1

[47] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):334–352, 2004. 1

[48] Rey Reza Wiyatno and Anqi Xu. Physical adversarial textures that fool visual object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4822–4831, 2019. 2, 6

[49] Christian Wojek, Stefan Walk, Stefan Roth, Konrad Schindler, and Bernt Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *IEEE transactions on pattern analysis and machine intelligence*, 35(4):882–897, 2012. 1

[50] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE TPAMI*, 37(9):1834–1848, 2015. 1, 2, 6

[51] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. 2

[52] Bin Yan, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 990–999, 2020. 2, 3, 6, 7

[53] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011. 5

[54] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. 2019. 2

[55] Zheng Zhu, Qiang Wang, Bo Li, Wu Wei, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pages 103–119, 2018. 2