Towards Universal Physical Attacks on Single Object Tracking

Li Ding[1,2*], **Yongwei Wang**[2*], **Kaiwen Yuan**[2], **Minyang Jiang**[2],
Ping Wang[1†], **Hua Huang**[3†], **Z. Jane Wang**[2]
[1]School of Information and Communications Engineering, Xi'an Jiaotong University,
[2]Department of Electrical and Computer Engineering, University of British Columbia,
[3]School of Artificial Intelligence, Beijing Normal University
{dinglijay,yongweiw,kaiwen, minyang, zjanew}@ece.ubc.ca, ping.fu@xjtu.edu.cn, huahuang@bnu.edu.cn

Abstract

Recent studies show that small perturbations in video frames could misguide single object trackers. However, such attacks have been mainly designed for digital-domain videos (i.e., perturbation on full images), which makes them practically infeasible to evaluate the adversarial vulnerability of trackers in real-world scenarios. Here we made the first step towards physically feasible adversarial attacks against visual tracking in real scenes with a universal patch to camouflage single object trackers. Fundamentally different from physical object detection, the essence of single object tracking lies in the feature matching between the search image and templates, and we therefore specially design the maximum textural discrepancy (MTD), a resolution-invariant and target location-independent feature de-matching loss. The MTD distills global textural information of the template and search images at hierarchical feature scales prior to performing feature attacks. Moreover, we evaluate two shape attacks, the regression dilation and shrinking, to generate stronger and more controllable attacks. Further, we employ a set of transformations to simulate diverse visual tracking scenes in the wild. Experimental results show the effectiveness of the physically feasible attacks on SiamMask and SiamRPN++ visual trackers both in digital and physical scenes.

Introduction

Single object tracking has attracted increasing attention in security related applications, such as autonomous driving, intelligent surveillance and human-machine interaction. The visual tracking task resorts to creating the dynamic correspondence (e.g., position) between a moving object in a given template frame and that in subsequent search frames without prior knowledge of object categories. Recently, there have been significant improvements in tracking performance with the adoption of deep convolutional neural networks (DNNs). Providing a good tradeoff between real-time tracking and accuracy, the Siamese-based trackers, e.g., SiamRPN (Li et al. 2018), SiamRPN++(Li et al. 2019), SiamMask(Wang et al. 2019), have become the mainstream approaches in visual tracking.

DNNs are shown vulnerable to adversarial perturbations, termed as adversarial attacks (Szegedy et al. 2014). Such attacks exist for different vision tasks implemented with DNNs, e.g., image classification (Kurakin, Goodfellow, and Bengio 2016), object detection (Xie et al. 2017) and visual tracking (Yan et al. 2020a). Generally, adversarial attacks can be categorized into digital attacks and physical attacks, depending on which domain to inject the perturbations (Huang et al. 2020). Specifically for single object tracking, recent studies primarily target at digital attacks (Yan et al. 2020a; Guo et al. 2019), leaving physical visual attacks rarely explored. Indeed, physical attacks are much more challenging than digital attacks due to practical constraints and feasibility.

In digital attacks, adversarial perturbations can be injected into any pixel of an image, and they can be different from image to image. In physical attacks, however, it requires the perturbation region to be small enough to be physically feasible, universal to diverse instances and robust to physical conditions (e.g., preprocessing, luminance factor). Also, physical attacks are more challenging to be detected and defended against, making them more threatening to trackers than digital attacks.

Despite certain pioneering explorations in physical attacks, existing works mainly focus on attacking image classifiers (Eykholt et al. 2018; Athalye et al. 2018) or object detectors (Chen et al. 2018; Huang et al. 2020). Probably against our intuition, though the task of visual tracking appears related with object detection (i.e., providing object bounding-box), their working mechanisms differ considerably. Object detection has one input and it estimates all locations of interested objects (instance-agnostic and category-dependent) while the single object tracking has two inputs and only localizes the user-specified target dynamically yet with no prior information of its category. In essence, visual tracking extracts and matches features between the template and search frames.

As illustrated in Fig. 1, it is challenging to attack the feature matching module. First, the dimensionality of feature maps are different between template and search frames. Therefore it is infeasible to employ the feature space adversarial loss proposed for classifiers (Inkawhich et al. 2019). Second, the coordinates of the tracking object are not spatially aligned in the feature space, which makes pixel-wise

[*]Equal contribution, alphabetical order.

[†]Corresponding Author.

feature comparison proposed for physical attacks on object detectors (Zhao et al. 2019) fail to generate effective perturbations in visual tracking. It is worth emphasizing that existing attack methods on classification and object detection cannot be applied on visual tracking. It necessitates a novel approach to *de-match* the Siamese features from the two branches.

Meanwhile, it is desirable that the adversaries can control the shape of the target's bounding-box predictions, i.e., misleading bounding boxes to dilate or shrink promptly yet consistently over time. Further, physical attacks demand practical considerations, e.g., a patch small enough to be physically feasible, universally valid to different instances within the same category, and robust to physical conditions and tracker re-initialization. In this work, we made the first attempt towards physically feasible universal attacks on SOTA Siamese-based visual trackers. The proposed method generates effective patches to significantly reduce the tracking performance of victim trackers in physically feasible scenarios. Overall, our main contributions are three-fold:

- *We present the first physically feasible attack approach to evaluate the adversarial vulnerability of SOTA Siamese-based visual trackers. Our attack is universal to different instances from the same category and robust in physical conditions. This work can be a baseline to evaluate the robustness of Siamese-based visual trackers in the wild.*
- *We propose the maximum textural discrepancy (MTD) loss function to misguide visual trackers by de-matching the template and search frames at hierarchical feature scales. Further, we consider the entire tracking pipeline, evaluating different shape attacks and optimization strategies to generate stronger and more controllable attacks.*
- *Experimental results show that the proposed physically feasible attacks can efficiently fool SiamMask and SiamRPN++ both in standard visual tracking datasets and in physical conditions. (Digital scenes are to imitate universal physical attacks in the digital domain.)*

Related Works

Siamese-based Visual Tracking

Single Object Tracking (SOT) aims to track an arbitrary object in an online video stream without knowing the object category in advance. Different from object detection, SOT requires the tracker capable of tracking any object with a one-shot glance. Generally, SOT can be formulated as a *similarity learning* problem. Since the seminar work in (Bertinetto et al. 2016) based on a fully-connected Siamese architecture (SiamFC), there has been increasing interest in SOT by leveraging the fast running speed and expressiveness power of deep neural networks.

A Siamese network consists of two identical φ branches, which transform an exemplar image z and a candidate image x to the feature space prior to fusing them to return a score. The SiamFC tracker (Bertinetto et al. 2016) first introduced a correlation layer which highly improved the tracking accuracy. SiamRPN (Li et al. 2018) formulated SOT as a local one-shot detection task. Then it explored the region proposal sub-network (RPN) (Ren et al. 2015) to yield faster speed and competitive tracking performance. To address the translation invariance issue, SiamRPN++ (Li et al. 2019) introduced a spatial-aware sampling strategy to significantly boost its performance gain by utilizing more sophisticated networks. SiamRPN++ also introduced the layerwise and depthwise aggregation module to further increase the tracker's performance. More recently, researchers studied the computational speed of visual trackers due to the pixel-level position estimate. SiamMask (Wang et al. 2019) alleviated this problem by formulating SOT as a multi-task learning problem. The SiamMask tracker involved training three tasks jointly, i.e., similarity matching module for dense response maps, RPN subnetwork for bounding box regression and binary segmentation for position refinement. SiamMask achieves the state-of-the-art performance on real-time visual tracking.

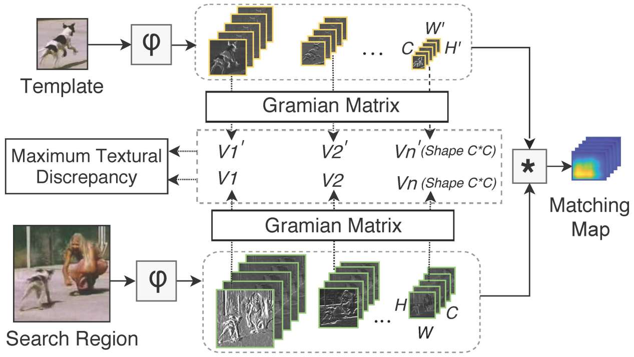

Figure 1: The framework of the Siamese-based matching network. Given a target template and the search region, their features are extracted by a Siamese network φ. Since the features could be in different shapes due to different image sizes of the template and search input, the features are matched using a cross-correlation layer to generate the matching map (here $*$ denotes the cross correlation).

Digital Attacks on Visual Trackers

It's demonstrated that DNNs are vulnerable to adversarial attacks on various computer vision tasks in the digital image domain, e.g., classification (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015; Xie et al. 2017), object detection and segmentation (Xie et al. 2017), or some recent explorations on visual trackers (Yan et al. 2020a,b). The work (Yan et al. 2020a) employed the generative adversarial networks (Goodfellow et al. 2014) with the proposed cooling-shrinking loss to generate imperceptible noise to attack the SiamRPN++ tracker. Then the perturbation was added to the template or search images on the network input (after pre-processing), which makes the attack unfeasible even in digital attacks. Guo et al. (Guo et al. 2019) proposed an online incremental attack. This attack exploits the spatial and temporal consistency in video frames so that the adversary fools object trackers with slight perturbations at each temporal frame. In work (Yan et al. 2020b), the authors

Attack	PF	SOTA	Universal	Re-initialization
(Guo et al. 2019)	×	✓	×	×
(Yan et al. 2020a)	×	✓	×	×
(Chen et al. 2020)	×	✓	×	×
(Yan et al. 2020b)	×	✓	×	×
(Wu et al. 2019)	×	✓	×	×
(Wiyatno and Xu 2019)	×	×	✓	×
Proposed Attacks	✓	✓	✓	✓

Table 1: Comparison of existing and the proposed adversarial attacks on visual trackers. “PF” denotes “Physically Feasible”; “SOTA” indicates the attacks are effective for SOTA visual trackers.

evaluated the vulnerability of Siamese-trackers by hijacking the bounding box of the tracking object to be a different shape or to targeted position. Chen et al. (Chen et al. 2020) generated the perturbation on the template frame with dual attention. Wu et al. (Wu et al. 2019) proposed 3D adversarial examples for visual trackers. Unfortunately, these attacks only work in the digital domain since adversarial perturbations were specifically designed differently for exemplar and search frames. This makes it infeasible to extend to physically realizable attacks.

One More Attack on Visual Trackers

In the interesting work (Wiyatno and Xu 2019), the authors designed an adversarial poster displayed on a big screen to fool the GOTURN tracker (Held, Thrun, and Savarese 2016) when a person approaches the screen. However, there are major differences between this work and our proposed attack. First, (Wiyatno and Xu 2019) requires the poster to be big enough to fully cover tracking objects – the poster size is 2.6m×2m while halving the size would fail to attack. Essentially, this is an extension of digital attacks since the perturbations can lie in arbitrary regions in the background. Second, the work only largely perturbs search images without changing the template image. This is unrealistic in physical conditions and it cannot handle model re-initialization in trackers. Third, the victim GOTURN tracker is obsolete which works differently from SOTA trackers. In contrast, we examine the tracking pipeline and propose novel loss functions for adversarial attacks. The proposed method generates portable adversarial patches, small yet effective to attack the state-of-the-art trackers. Since our patches appear both in template and search images, they are capable of fooling trackers even with model re-initialization. We focus on more practical physical attacks on SOTA trackers. Such physically feasible attacks are more dangerous yet less explored. In Table 1, we compare the proposed methods with existing attacks from different aspects.

Physically Feasible Attacks

In this section, we present the pipeline of the proposed method, as shown in Fig. 2. We first formulate the problem of physically feasible attacks on visual tracker, and we then elaborate our method in detail.

For a Siamese tracker with the matching network φ (see Fig. 1), we denote the template image as $\boldsymbol{z}^{(t)} \in \mathbb{R}^{w_z \times h_z \times c}$ at the t-th re-initialization, and the search image as $\boldsymbol{x}^{(t,s)} \in \mathbb{R}^{w_x \times h_x \times c}$ at the s-th frame corresponding to the t-th trial. The search image passes through a sub-window ω which involves cropping, padding and resizing operations before it is fed into the matching network. We denote the extracted features from the template and search images as $\varphi(\boldsymbol{z}^{(t)}) \in \mathbb{R}^{w'_z \times h'_z \times c'}$, $\varphi(\omega(\boldsymbol{x}^{(t,s)})) \in \mathbb{R}^{w'_x \times h'_x \times c'}$, respectively.

In physically feasible tracking attacks, adversaries attempt to find a universal patch $\boldsymbol{\delta}$ to significantly degrade the performance of the visual trackers *over time*. Suppose a target has been camouflaged by the adversarial patch, let us denote the exemplar image as $\boldsymbol{z}_{\delta}^{(t)}$ and the search image as $\boldsymbol{x}_{\delta}^{(t,s)}$, respectively. $\boldsymbol{z}_{\delta}^{(t)}$ and $\boldsymbol{x}_{\delta}^{(t,s)}$ can be expressed as,

$$\begin{aligned} \boldsymbol{z}_{\delta}^{(t)} &= \boldsymbol{z}^{(t)} \odot \boldsymbol{M}^{(t)} + (\boldsymbol{I}^{(t)} - \boldsymbol{M}^{(t)}) \odot \boldsymbol{\delta} \\ \boldsymbol{x}_{\delta}^{(t,s)} &= \boldsymbol{x}^{(t,s)} \odot \boldsymbol{M}^{(t,s)} + (\boldsymbol{I}^{(t,s)} - \boldsymbol{M}^{(t,s)}) \odot \boldsymbol{\delta} \end{aligned} \tag{1}$$

where $\odot$ represents the element-wise Hadmard product; $\boldsymbol{M}^{(t)}, \boldsymbol{M}^{(t,s)}$ denote binary masks for $\boldsymbol{z}_{\delta}^{(t)}$ and $\boldsymbol{x}_{\delta}^{(t,s)}$, respectively; $\boldsymbol{I}^{(t)}, \boldsymbol{I}^{(t,s)}$ represent all-one matrices with the same dimension as $\boldsymbol{M}^{(t)}, \boldsymbol{M}^{(t,s)}$, respectively.

Similar to existing digital attacks (Yan et al. 2020a; Guo et al. 2019), we will blind the Siamese-based visual trackers *over time*. Concretely, assume that a victim tracker is re-initialized with an exemplar image $\boldsymbol{z}_{\delta}^{(t)} \in \boldsymbol{\mathcal{Z}}_a$ where an adversarial patch has been attached on the tracking object. Correspondingly, the search frames are $\boldsymbol{x}_{\delta}^{(t,s)} \in \boldsymbol{\mathcal{X}}_a^{(t)} \triangleq \{\boldsymbol{x}_{\delta}^{(t,1)}, \boldsymbol{x}_{\delta}^{(t,2)}, \cdots, \boldsymbol{x}_{\delta}^{(t,S)}\}$. Then with $\boldsymbol{\delta}$, the tracker will fail to correctly match $\boldsymbol{z}_{\delta}^{(t)}$ with $\boldsymbol{x}_{\delta}^{(t,s)},\ s = 1, \cdots, S$.

Maximum Textural Discrepancy

As illustrated in Fig. 1, the essence of Siamese-trackers is to locate the target in search regions via a matched filter represented by φ. The activated matched features will then be delivered to downstream functional sub-networks. Therefore, a sufficient condition to blind a Siamese-tracker is to de-match its upstream representations.

The work (Inkawhich et al. 2019) proposed a targeted-classification attack by minimizing the representation distance between the source and target images in the feature space. This attack was shown to achieve high transferability for classification tasks. However, unlike the targeted classification attack, there is no “target image” (i.e., an instance of a targeted class) in visual tracking. Also, the dimensionality difference of $\varphi(\boldsymbol{z}^{(t)})$ and $\varphi(\boldsymbol{x}^{(t,s)})$ hinders the direct calculation of the feature distance.

Recent studies reveal an intriguing phenomenon that neural networks are biased towards *textures* in image classification (Geirhos et al. 2019; Zhang and Zhu 2019). Textures refer to certain spatially stationary statistics in natural images, which can be calculated from the Gramian matrix in the feature space (Gatys, Ecker, and Bethge 2016; Johnson, Alahi, and Fei-Fei 2016). The textural feature is independent of feature dimensionality and it also explicitly exploits the vulnerability of neural networks.

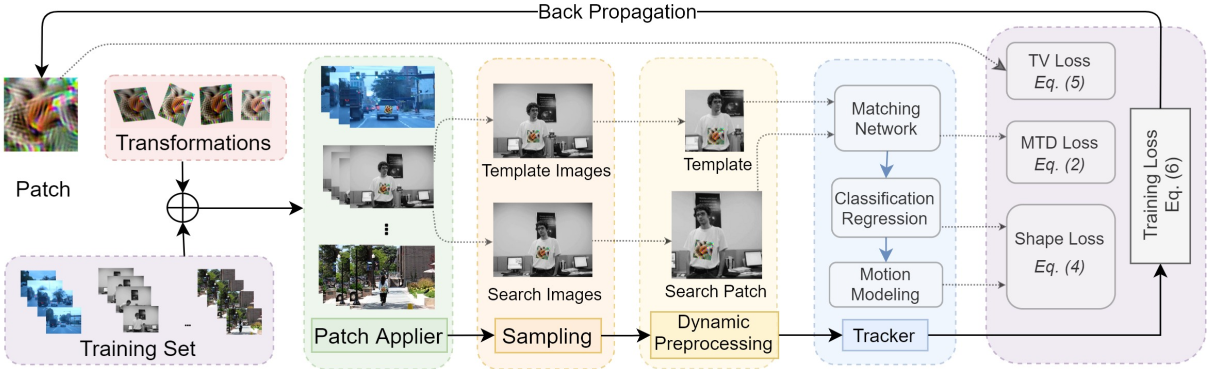

Figure 2: Overview of the proposed attack pipeline. Given a randomly-initialized patch and the training streams, firstly the patch is transformed randomly (e.g., random change in brightness, contrast, color, translations, rotation, sheering etc). Then the *patch applier* overlays the patch onto the target. At each iteration, we sample a random batch of frames, dynamically processing and passing them into the victim trackers. Finally the patch is updated by minimizing the proposed overall loss function.

Motivated by this, we propose the *Maximum Textural Discrepancy* (MTD) as a novel loss function to fool the matched filter. Specifically, the objective is to maximize the discrepancy of textural representations between $\boldsymbol{z}_{\delta}^{(t)}$ and $\boldsymbol{x}_{\delta}^{(t,s)}$ so that the feature representations are de-matched in the upstream of visual trackers. The hierarchical MTD loss on D layers, $\mathcal{L}_{MTD}$, is defined as:

$$\mathcal{L}_{MTD}(\boldsymbol{z}_{\delta}^{(t)}, \boldsymbol{x}_{\delta}^{(t,s)}) = -\frac{1}{D} \sum_{d \in D} \left|\left| \boldsymbol{\mathcal{G}}\left(\varphi_d(\boldsymbol{z}_{\delta}^{(t)})\right) - \boldsymbol{\mathcal{G}}\left(\varphi_d(\omega(\boldsymbol{x}_{\delta}^{(t,s)}))\right) \right|\right|_{\mathcal{F}} \tag{2}$$

where $\mathcal{F}$ denotes the Frobenius norm; $\boldsymbol{\mathcal{G}}$ represents the Gramian matrix operator $\boldsymbol{\mathcal{G}} : \boldsymbol{f}^{(d)} \in \mathbb{R}^{w_d \times h_d \times c_d} \mapsto \mathbb{R}^{c_d \times c_d}$, where $\boldsymbol{f}^{(d)} \triangleq \{\boldsymbol{f}_{(1)}^{(d)}, \cdots, \boldsymbol{f}_{(c^d)}^{(d)}\}$ denotes the feature map profile composed of feature map $\boldsymbol{f}_{(i)}^{(d)} \in \mathbb{R}^{w_d \times h_d}$ of the i-th channel ($i = 1, \cdots, c_d$), at the d-th layer ($d = 1, 2, \cdots, D$). Concretely, given two feature maps $\boldsymbol{f}_{(i)}^{(d)}, \boldsymbol{f}_{(j)}^{(d)} \in \mathbb{R}^{w_d \times h_d}$ from the feature map profile $\boldsymbol{f}^{(d)}$, the Gramian output $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{f}^{(d)})$ at the i,j-th component ($i, j = 1, \cdots, c_d$) can be computed:

$$\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{f}^{(d)}) = \left\langle Vec(\boldsymbol{f}_{(i)}^{(d)}),\ Vec(\boldsymbol{f}_{(j)}^{(d)}) \right\rangle \tag{3}$$

where $<, >$ and $Vec(\cdot)$ denote the inner product and matrix vectorization operation, respectively.

Proposition 1. *The Gramian matrix in Eq. (3) turns out to be the correlation matrix of feature maps from different channels. By maximizing the textural discrepancy measured by the Gramian matrix, we can minimize the correlation between* $\boldsymbol{z}_{\delta}^{(t)}$ *and* $\boldsymbol{x}_{\delta}^{(t,s)}$ $(\forall t, s)$ *in the feature space*. Please find our proof in Appendix A.

Remark. From the analysis above, we conclude that the MTD loss (in Eq. (2)) explicitly de-matches the feature representations produced from the matched filter φ.

Shape Attacks

In visual attacks, it is desirable that attackers can misguide bounding-box predictions promptly and consistently *over time*. Here we consider shape attacks (i.e., shape dilation or shrinking) by fooling the downstream regression subnetwork to make visual attacks in a controllable manner.

The SOTA Siamese trackers, e.g. SiamRPN++ and SiamMask, use RPN to locate the object's position, which consists of two branches: the regression network for proposal regression and the classification network for target or background prediction. The regression network predicts the shape of bounding boxes $\{(\widetilde{x}_i^{(s,t)}, \widetilde{y}_i^{(s,t)}, \widetilde{h}_i^{(s,t)}, \widetilde{w}_i^{(s,t)})\}_{i=1}^{N}$. The classification network discriminates the target from its background with the classification feature map and generates the similarity map. Further, motion modeling is adopted to re-rank the proposals' score $\{\widetilde{p}_i^{(s,t)}\}_{i=1}^{N}$. Finally bounding box with the highest score is selected as the target position.

In SiamMask and SiamRPN++, the motion model penalizes the position prediction and encourages the output to be spatially stable. Therefore it is challenging to interfere the final classification which could misguide trackers. As a result, alternatively we propose shape attacks by distracting the shape of the bounding boxes. In shape attacks, firstly we select a set of bounding boxes which provide top-K penalized scores: $\{\widetilde{p}_k^{(s,t)}\}_{k=1}^{K}$. Based on these penalized scores, we can explicitly consider the motion model in visual tracking. Concretely, the selected bounding boxes form a set $\Omega^{(s,t)} = \{(\widetilde{h}_1^{(s,t)}, \widetilde{w}_1^{(s,t)}), (\widetilde{h}_2^{(s,t)}, \widetilde{w}_2^{(s,t)}), \cdots, (\widetilde{h}_K^{(s,t)}, \widetilde{w}_K^{(s,t)})\}$. Denote the targeted bounding box shape as $(\overset{\vee}{h}, \overset{\vee}{w})$ and the regression margin as m_{τ}. The loss for regression shape attacks $\mathcal{L}_{Sha}$ can be written as:

$$\mathcal{L}_{Sha}(\boldsymbol{z}_{\delta}^{(t)}, \boldsymbol{x}_{\delta}^{(t,s)}) = \frac{1}{K} \sum_{k=1}^{K} max\left(\left| \widetilde{h}_k^{(s,t)} - \overset{\vee}{h} \right|_1 + \left| \widetilde{w}_k^{(s,t)} - \overset{\vee}{w} \right|_1,\ m_{\tau} \right) \tag{4}$$

In Eq.(4), with specified parameters ($\overset{\vee}{h}, \overset{\vee}{w}$ and m_{τ}), adversaries can control the desired shape of predicted bounding boxes after attack, i.e., shape dilation ($\overset{\vee}{h} = \overset{\vee}{w} = 1$) or shape shrinking ($\overset{\vee}{h} = \overset{\vee}{w} = -1$) attacks. It is worthy to mention that the proposed Shape loss is distinct from that in (Yan et al. 2020a), because their loss necessitates a clean video as the input; however, we do not have such information in the physically feasible scenes. Moreover, we incorporate into our loss formulation the motion model in tracking attacks.

Universal Physical Attacks

Practical physical attacks present more challenges than digital attacks on the trackers. As elaborated below, we address three challenges: (1) physically realizable; (2) universal to diverse instances; and (3) robust to physical conditions and tracker re-initialization.

Physically feasible. The proposed patch-based loss functions (i.e. Eqs.(2) and (4)) can be directly applied to the physical conditions. Since natural images (or patches) generally look smooth, we consider the smoothness constraint to avoid sharp texture transitions and increase its "stealthiness". We use the total variation (TV) (Sharif et al. 2016; Thys, Van Ranst, and Goedemé 2019) to penalize the smoothness term,

$$\mathcal{L}_{TV}(\boldsymbol{z}_{\delta}^{(t)}, \boldsymbol{x}_{\delta}^{(t,s)}) = \frac{1}{w_{\delta}\ h_{\delta}} \sum_{i=1}^{w_{\delta}} \sum_{j=1}^{h_{\delta}} \left\{ \left| \boldsymbol{\delta}_{i+1,j} - \boldsymbol{\delta}_{i,j} \right|^2 + \right. \\ \left. \left| \boldsymbol{\delta}_{i,j+1} - \boldsymbol{\delta}_{i,j} \right|^2 \right\}^{1/2} \tag{5}$$

where w_{δ}, h_{δ} represent the width and height of the adversarial patch $\boldsymbol{\delta}$, respectively.

Universality. *Universality* could mean two aspects in physical attacks. First, the patch is effective for different instances within the same category (e.g. human, cars). Second, the patch remains adversarial for instances from different categories. Here we focus on the first case, and we leave the latter scenario as future work. Given the randomly sampled exemplar image $\boldsymbol{z}_{\delta}^{(t)} \in \boldsymbol{\mathcal{Z}}_a$ and search frame $\boldsymbol{x}_{\delta}^{(t,s)} \in \boldsymbol{\mathcal{X}}_a^{(t)} \triangleq \{\boldsymbol{x}_{\delta}^{(t,1)}, \boldsymbol{x}_{\delta}^{(t,2)}, \cdots, \boldsymbol{x}_{\delta}^{(t,S)}\}$, the overall objective function $\mathcal{L}$ for universal physical attacks becomes,

$$\mathcal{L}(\boldsymbol{z}_{\delta}, \boldsymbol{x}_{\delta}) = \sum_{\boldsymbol{z}_{\delta}^{(t)} \in \boldsymbol{\mathcal{Z}}_a} \sum_{\boldsymbol{x}_{\delta}^{(t,s)} \in \boldsymbol{\mathcal{X}}_a^{(t)}} \alpha \mathcal{L}_{MTD} + \beta \mathcal{L}_{Sha} + \gamma \mathcal{L}_{TV} \tag{6}$$

where α, β, γ denote the weights for loss functions $\mathcal{L}_{MTD}$, $\mathcal{L}_{Sha}$ and $\mathcal{L}_{TV}$, respectively.

Robustness. Robustness is important to ensure that the attacks work properly in the physical world, where the patch may suffer from different visual distortions when captured by a visual tracker (e.g. camera from a moving car). To mimic the real world conditions, we include diverse transformations and apply the expectation over transformation (EoT) (Athalye et al. 2018) on adversarial patches. Apart from some affine transforms (e.g. rotation, translation) in (Athalye et al. 2018), we also consider changes in perspectives, brightness, contrast and color jittering. Detailed setting can be found in Appendix B. Denote the transformation as $\mathcal{T}$. Our robust adversarial patch on visual tracking $\boldsymbol{\delta}^{UPS}$ can be obtained by,

Algorithm 1: The proposed algorithm of universal and physically feasible attacks on visual tracking.

```
Data: Training video streams set $\boldsymbol{\mathcal{X}}$, number of template
      images $T$, number of search images $S$ at one
      template, regression margin $m_{\tau}$, loss weights
      $\alpha, \beta, \gamma$, maximum number of iterations $M$.
Result: Optimized adversarial patch $\boldsymbol{\delta}^{UPS}$.
Initialize adversarial patch with Gaussian noise, $iter = 0$;
while $iter < M$ do
    Sample a template image $\boldsymbol{z}^{(t)}$ from $\boldsymbol{\mathcal{X}}$;
    Sample one search image $\boldsymbol{x}^{(t,s)}$ from $\boldsymbol{\mathcal{X}}$ from nearby
      frames of $\boldsymbol{z}^{(t)}$;
    Sample a transform $\mathcal{T}$ on patch $\boldsymbol{\delta}$, warp transformed
      patch $\boldsymbol{\delta}$ to template $\boldsymbol{z}^{(t)}$;
    Sample a transform $\mathcal{T}$ on patch $\boldsymbol{\delta}$, warp transformed
      patch $\boldsymbol{\delta}$ to search image $\boldsymbol{x}^{(t,s)}$;
    Pre-process the image pair $\{\boldsymbol{z}^{(t)}, \boldsymbol{x}^{(t,s)}\}$ and input
      them to victim tracker;
    Compute MTD loss $\mathcal{L}_{MTD}$ from Eq.(2);
    Select bounding boxes set
      $\Omega^{(s,t)} = \{(\widetilde{h}_1^{(s,t)}, \widetilde{w}_1^{(s,t)}), \cdots, (\widetilde{h}_K^{(s,t)}, \widetilde{w}_K^{(s,t)})\}$
      based on top-$K$ penalized scores;
    Compute the shape loss $\mathcal{L}_{Sha}$ from Eq.(4);
    Compute the total variation loss $\mathcal{L}_{TV}$ from Eq.(5);
    Compute the overall loss $\mathcal{L}(\boldsymbol{z}_{\delta}, \boldsymbol{x}_{\delta})$ from Eq.(6);
    Optimize $\boldsymbol{\delta}$ using the Adam optimizer from Eq.(7);
    $iter \leftarrow iter + 1$;
end
```

$$\boldsymbol{\delta}^{UPS} = \arg\min_{\boldsymbol{\delta}} \ \mathbb{E}_{\boldsymbol{\delta} \sim \mathcal{T}\boldsymbol{\delta}} \Big[\mathcal{L}(\boldsymbol{z}_{\delta}, \boldsymbol{x}_{\delta}) \Big] \tag{7}$$

where $\mathcal{L}(\boldsymbol{z}_{\delta}, \boldsymbol{x}_{\delta})$ is given in Eq. (6).

The overall pipeline of the proposed attack, the universal physically feasible attack, is described in Algorithm 1.

Experiments

In this section, we empirically evaluate the effectiveness of the proposed attacks on visual tracking both in digital and physically feasible scenes. The attacks in digital scenes are to imitate the physically feasible attacks in the real world. Therefore we can quantitatively assess our attacks on the standard datasets and tune parameters more efficiently. The experiments were conducted on one NVIDIA RTX-2080 Ti GPU card using PyTorch (Paszke et al. 2019).

Experimental Setup

In all experiments, we keep the patch and object size ratio within 20% to be physically feasible. For parameters in the overall loss expression in Eq.(6), we set $D = 3$, and the loss weights are set respectively as: $\alpha = 1000, \beta = 1, \gamma = 0.1$. In the Shape loss in Eq.(4), we set $K = 20$. More concretely, for the shrinking attack, we set $\overset{\vee}{h} = -1, \overset{\vee}{w} = -1, m_{\tau} = 0.7$;

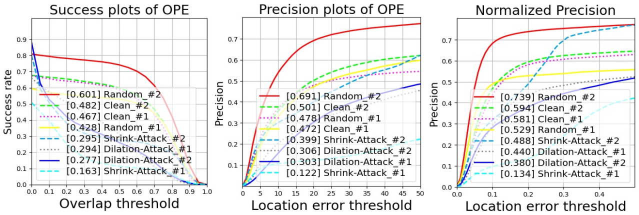

Figure 3: Quantitative comparison of three metrics on *person* with different thresholds.

and for the dilation attack, we use $\overset{\vee}{h} = 1, \overset{\vee}{w} = 1, m_\tau = 0.7$. Please refer to Appendix B for more detailed settings.

The victim models are SOTA Siamese-based trackers: SiamMask and SiamRPN++ (Wang et al. 2019; Li et al. 2019). Adversarial patches were trained and tested on different instances and background to better evaluate their generalization and robustness.

To quantitatively evaluate the attack performance, we employ three popular metrics in visual tracking: *success*, *precision* and *normalized precision* (Muller et al. 2018; Fan et al. 2019). The *success* is computed as the *Intersection-over-Union* (IoU) between the predicted bounding box and the groundtruth. The *precision* is measured by the distance between the tracking result and groundtruth bounding box in pixels. The *normalized precision* is computed with the Area Under Curve (AUC) between 0 and 0.5 (Muller et al. 2018).

Physically Feasible Attacks in Digital Scenes

For the physically feasible attacks in digital scenes, we experimented on three object categories: *person*, *car* and *cup* from the Large-scale Single Object Tracking (LaSOT) dataset (Fan et al. 2019). Each category consists of 20 videos, among which we randomly select one video for adversarial patch generation. We then attack the rest 19 videos within the same category by warping the patch on the target.

In Table 2, we report the performance drop on both trackers (Wang et al. 2019) where we consider the white-box attacks individually. As a comparison experiment, we also evaluate the influence of random patches (i.e. without training) with the same patch/object ratio. Interestingly, we observe that random patches can even boost the tracking performance. The reason might be that the random patch essentially provides more useful information for target localization. By contrast, there is a sharp performance decrease with adversarial patches in each category. We also quantitatively compare three metrics with different thresholds in Fig. 3 on “person”. Clearly, on SiamMask and SiamRPN++, adversaries can significantly reduce the tracking performance with our generated patches while the non-trained random patches improve the tracking performance. We have the same observations on “car” and “bottle” categories.

In Fig.4, we show visual examples of the “dilation” and “shrinking” attacks on the “person” object on SiamRPN++. There are two observations: (1) the IoU (2nd row) of both attacks quickly drop to a low value with an adversarial patch; otherwise, the IoU keeps a high value without attacks. Correspondingly, SiamRPN++ produces dilated (1st row) or shrinked prediction boxes (3rd row). (2) The tracker keeps losing the target even if we re-initialize it with a new template image (2nd row). These observations further confirm the effectiveness of the proposed physically feasible attacks.

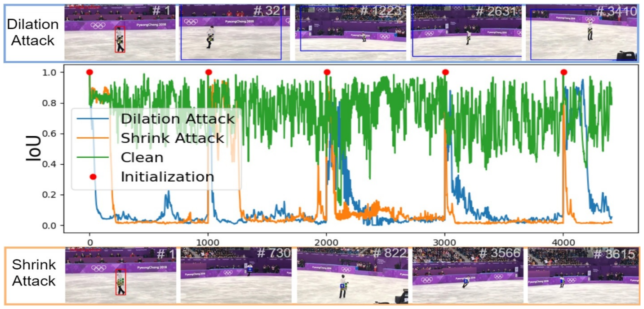

Figure 4: Illustration of the effectiveness of the generated patch. The 1st and 3rd rows show visual examples of the proposed dilation and shrinking attacks on “person”. Bounding boxes in red depict the initialization positions while the blue ones display predicted positions after our attacks. The 2nd row shows the comparison in IoU prediction between clean and attacks over time. The red dot indicates model initialization at that time.

Physically Feasible Attacks in Real Scenes

After having verified our attacks in virtual scenes, we conduct experiments to demonstrate their efficacy in real world environments. In physical attacks, we mainly experiment on the “person” and “bottle” categories with diverse instances.

In Fig. 5, we show example frames of tracking results after physical attacks with “dilation” (rows 1 - 3) and “shrinking” attacks (rows 4 - 6), respectively. In “dilation” attacks, the predicted bounding box dilates to the full frame size which fails the victim tracker gradually yet promptly. Specifically, in the second row, the template is selected as the white bottle (i.e. Frame #1 in red as the target, textural region in the middle as our patch). In less than one second (*Frame #24*, real time=0.8 second), however, the tracker has been confused by erroneously tracking two other bottles as the target. The bounding box continues dilating until it “fills in” the whole image frame (*Frames #24 – 297*). At Frame #298, we re-initialize the tracker with the target object, however again the tracker has been easily fooled by our patch on it. In the third row, we display a small patch on a mobile phone screen (screen size 14.9 cm × 7.1 cm), the tracker quickly gets misguided even with model re-initialization (*Frames #70 – 121*). By contrast, when we remove the patch, the tracker can track well (*Frames #423 – 494*). Conversely, in “shrinking” attacks (rows 4 – 6), the predicted bounding box quickly shrinks to a small region and produces unstable predictions which eventually fails in tracking the target. For example, the tracker may confuse itself with objects near the target (*Frames #220 – 460* in row 4; *Frames #62 – 286* in row 6). The tracking predictions may also fall onto the patch and the predicted bounding boxes could be “threw away”

Category	Metric	Random (↓%)		Dilation Attack (↓%)		Shrinking Attack (↓%)	
		#1	#2	#1	#2	#1	#2
Person	Success	-24.7	8.4	42.5	37.0	38.8	65.1
	Precision	-37.9	-1.3	38.9	35.8	20.4	74.2
	Norm Precision	-24.4	9.0	36.0	24.3	17.8	76.9
Car	Success	-11.9	-2.7	46.9	57.5	32.2	44.5
	Precision	-11.1	-5.3	39.5	41.2	21.2	42.8
	Norm Precision	-12.6	-3.0	45.2	41.7	19.8	43.0
Bottle	Success	-9.9	-16.5	49.5	38.2	45.7	25.4
	Precision	-14.9	-30.3	69.5	67.9	18.5	20.5
	Norm Precision	-12.9	-22.9	44.1	27.6	5.6	34.0

Table 2: Quantitative performance evaluation of the proposed attacks on SiamMask (#1) and SiamRPN++ (#2) with *person*, *car* and *bottle* categories. The table reports the percentage of performance drop in tracking with patches from: *Random, Dilation* and *Shrinking* attacks, respectively. "↓" denotes performance drop and larger values are preferred.

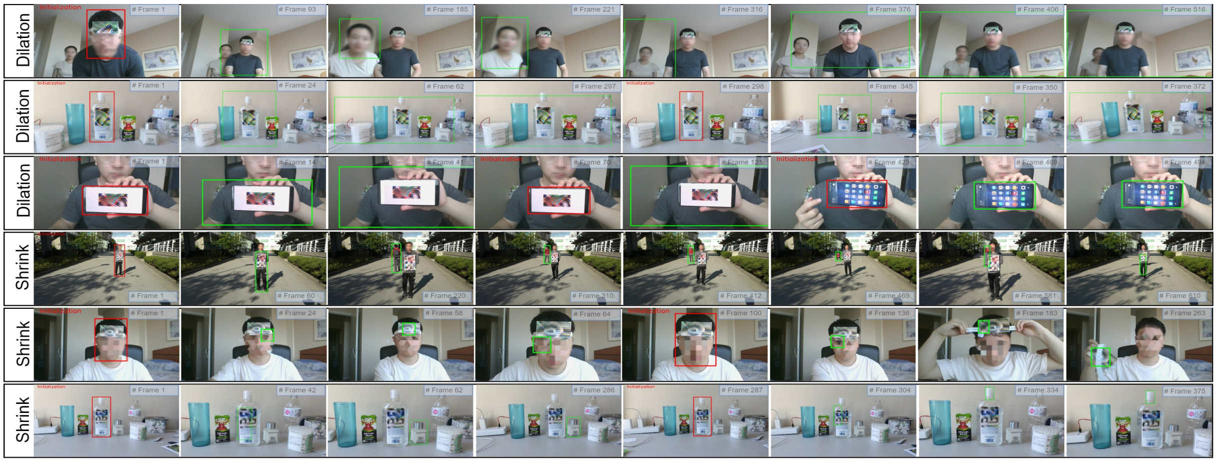

Figure 5: Example frames of tracking results of the proposed physically feasible attacks in real scenes.

Category	Success (↓%)	Precision (↓%)	Norm Precision (↓%)
Person	65.6	36.0	54.1
Bottle	83.5	77.5	91.7

Table 3: Quantitative performance evaluation in physical attacks. The symbol "↓" denotes performance drop and larger values indicate stronger attacks.

Metric	Dilation Attack (↓%)		Shrink Attack (↓%)	
	w/o MTD	w/ MTD	w/o MTD	w/ MTD
Success	28.5	**37.0**	53.7	**65.1**
Precision	26.5	**35.8**	55.5	**74.2**
Norm Precision	18.9	**24.3**	62.7	**76.9**

Table 4: Ablation study of the MTD loss on SiamMask. "↓" denotes performance drop and larger values are preferred.

intentionally (*Frames #136 – 263* in row 5).

We also quantitatively measure the performance of physical attacks. We manually annotate target objects on "bottle" (row 2) and "person" (row 5). To approximately measure the performance on clean objects (without patch), we manually annotate the patch region and replace patch values with uniform intensity as 127. The performance drop of metrics are reported in Table 3.

Ablation Studies

We evaluate the influence of the MTD loss and patch size ratios. Ablation studies were conducted on the LaSOT dataset.

MTD loss function. We compare the performance drop (i.e. w/ and w/o the MTD loss) on the "person" object on SiamMask in Table 4. Clearly, the MTD loss can significantly boost the attack performance on three metrics. Similar observations for SiamRPN++ are shown in Appendix C. This observation implies that MTD loss indeed enhances the attack ability.

Patch size ratio. The patch size ratio is an important parameter in physically feasible attacks. Therefore, we evaluate the attack performance wrt different patch size ratios on SiamMask (#1) and SiamRPN++ (#2) in Fig. 6. In general, as the patch ratio increases from 15% to 35%, all three metrics decrease gradually, indicating stronger attack abilities. Therefore, the reported attack performances (with patch ratio as 20%) can be further improved if we utilize a larger patch size ratio in the experiments.

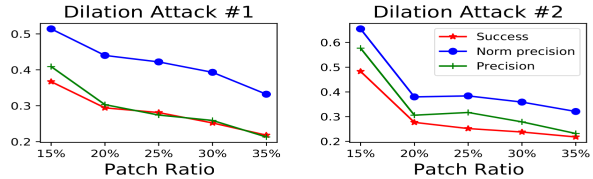

Figure 6: Attack performances as a function of the patch ratio.

Conclusion

As the first attempt, we study universal physically feasible attacks against single object tracking. To generate an effective patch, we propose the MTD loss to effectively de-match the template and search frames in hierarchical feature levels. We then propose two shape attacks to misguide visual trackers in a more controllable way. Finally we evaluate different optimization strategies to make the patch universal to different instances within a category and more robust to practical environments. Experimental results demonstrate that the proposed methods can significantly degrade advanced visual trackers' performances in the physically feasible attack setting. Our exploration on physically feasible attacks raises the security concerns of real-world visual tracking.

Acknowledgments

We acknowledge financial support from the National Natural Science Foundation of China (NSFC) under Grant No. 61936011, and the Natural Sciences and Engineering Research Council of Canada (NSERC).

Appendices

A. Proof of Proposition 1

Proof. For the template frame, denote its feature map profile $\boldsymbol{f} \triangleq \{\boldsymbol{f}_{(1)}, \boldsymbol{f}_{(2)}, \cdots, \boldsymbol{f}_{(c)}\}$ where c represents the number of channels produced by a certain layer of interest. Assume the feature maps can be modeled by multivariate random variables $\boldsymbol{r} \triangleq \left(r_{(1)}, r_{(2)}, \cdots, r_{(c)}\right)^T$. Denote the observations of a random variable $r_{(k)}$ by $\boldsymbol{v}_{(k)} \in \mathbb{R}^{wh}$ $(k = 1, \cdots, c)$, where $\boldsymbol{v}_{(k)}$ is the vectorized representation of the k-th feature map $\boldsymbol{f}_{(k)}$. For the search frame, we follow the same assumption in the feature layer and denote the random variables by $\boldsymbol{r}' \triangleq \left(r'_{(1)}, r'_{(2)}, \cdots, r'_{(c)}\right)^T$. We also denote the observations of a random variable $r'_{(k)}$ by $\boldsymbol{v}'_{(k)} \in \mathbb{R}^{w'h'}$, where $\boldsymbol{v}'_{(k)}$ is the vectorized representation of the k-th feature map $\boldsymbol{f}'_{(k)}$ $(k = 1, \cdots, c)$.

The i, j-th component $(i, j = 1, \cdots, c)$ of the Gramian operator $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{v})$ and $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{v}')$ can be computed:

$$\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j} = \boldsymbol{v}^T_{(i)} \cdot \boldsymbol{v}_{(i)}$$

$$\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,j} = \boldsymbol{v}'^T_{(i)} \cdot \boldsymbol{v}'_{(i)}$$

The textural discrepancy term described by $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{v})$ and $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{v}')$ can be expressed:

$$\begin{aligned}
\mathcal{L}_{td} &= \left|\left|\boldsymbol{\mathcal{G}}\left(\boldsymbol{v}\right) - \boldsymbol{\mathcal{G}}\left(\boldsymbol{v}\right)\right|\right|^2_{\mathcal{F}} \\
&= \sum_{i,j} \left(\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j} - \boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,j}\right)^2 \\
&= \sum_{i,j} \left[(\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j})^2 + (\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j})^2\right] \\
&\quad - 2\sum_{i \neq j} \boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j}\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,j} - 2\sum_{i} \boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,i}\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,i} \\
&= \sum_{i,j} \left[(\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j})^2 + (\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,j})^2\right] \\
&\quad - 2\sum_{i \neq j} \boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j}\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,j} - 2\sum_{i} \left|\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,i}\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,i}\right|
\end{aligned}$$

Clearly, maximizing $\mathcal{L}_{td}$ involves minimizing the inner product of diagonal components from two Gramian operator outputs.

Let us zero-pad $\boldsymbol{v}_{(k)} \in \mathbb{R}^{wh}$ and it yields $\widetilde{\boldsymbol{v}}_{(k)} \in \mathbb{R}^{w'h'}$, where we have $\widetilde{\boldsymbol{v}}^T_{(i)}\widetilde{\boldsymbol{v}}_{(j)} = \boldsymbol{v}^T_{(i)}\boldsymbol{v}_{(j)}$. Then,

$$\begin{aligned}
\sum_{i} \left|\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,i}\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,i}\right| &= \sum_{i} \left|\widetilde{\boldsymbol{v}}^T_{(i)} \cdot \widetilde{\boldsymbol{v}}_{(i)} \cdot \boldsymbol{v}'^T_{(i)} \cdot \boldsymbol{v}'_{(i)}\right| \\
&= \sum_{i} ||\widetilde{\boldsymbol{v}}_{(i)}||^2 \cdot ||\boldsymbol{v}'_{(i)}||^2 \\
&\geq \sum_{i} ||\widetilde{\boldsymbol{v}}^T_{(i)} \cdot \widetilde{\boldsymbol{v}}'_{(i)}||^2
\end{aligned}$$

Denote the cross-correlation between r.v. $r_{(i)}, r'_{(i)}$ as $\boldsymbol{K}_{r_{(i)}, r'_{(i)}}$, then we have:

$$\lim_{w'h' \to \infty} \widetilde{\boldsymbol{v}}^T_{(i)} \cdot \widetilde{\boldsymbol{v}}'_{(i)} = (w'h')^2 \cdot \boldsymbol{K}_{r_{(i)}, r'_{(i)}}$$

Therefore, given sufficient observations ($w'h' \to \infty$), we have:

$$\begin{aligned}
\mathcal{L}_{td} \leq &\sum_{i,j} \left[(\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j})^2 + (\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j})^2\right] \\
&- 2\sum_{i \neq j} \boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j}\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,j} - 2(w'h')^4 \sum_{i} |\boldsymbol{K}_{r_{(i)}, r'_{(i)}}|^2
\end{aligned}$$

Therefore, the textural discrepancy function $\mathcal{L}_{td}$ is a lower bound of the expression to the right represented by $|\boldsymbol{K}_{r_{(i)}, r'_{(i)}}|$. By maximizing $\mathcal{L}_{td}$, we can minimize the absolute value of the cross-correlation between feature representations of template and search frames modeled by multivariate r.v. $r_{(i)}$ and $r'_{(i)}$ for $i = 1, 2, \cdots, c$.

B. Implementation Details

We provide more details of the parameter setting in our experiments. We employ the Adam optimizer from the PyTorch platform with hyperparameters: exponential decays $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate $lr = 10$ (for intensity between [0,255]), weight decay set as 0, the batchsize set as 20, and the maximum training epochs $M = 300$.

Transforms	Parameters	Remark
brightness	0.1	brightness factor chosen uniformly from [0.9, 1.1]
contrast	0.1	contrast factor chosen uniformly from [0.9, 1.1]
hue	0.1	hue factor chosen uniformly from [-0.1, 0.1]
saturation	0.01	saturation factor chosen uniformly from [0.99, 1.01]
rotation	5	range of degrees chosen uniformly from [-5, 5]
translation	0.02	maximum absolute fraction (wrt image size) for translations
scaling	0.02	scale factor (wrt image size) chosen uniformly from [0.98, 1.02]
shearing	5	range of degrees chosen from [-5, 5]

Table 5: Patch transforms and parameters in the experiments.

In the MTD loss in Eq.(2), we choose $D = 3$. Specifically, for SiamMask and SiamRPN++ trackers, which utilize the ResNet-50 as the backbone network, D layers are the last three residual blocks for multi-scale feature extraction.

In the Shape loss in Eq.(4), we set $K = 20$. More concretely, for the shrinking attack, we set $\overset{\vee}{h} = -1, \overset{\vee}{w} = -1, m_\tau = 0.7$; and for the dilation attack, we use $\overset{\vee}{h} = 1, \overset{\vee}{w} = 1, m_\tau = 0.7$.

In the overall loss in Eq.(6), the loss weights are set respectively as: $\alpha = 1000, \beta = 1, \gamma = 0.1$.

In the final loss in Eq.(7), the transforms that we employed and their parameters have been listed in Table 5.

To generate adversarial patches, firstly we randomly select one video from a category (e.g. person) as the training data and create the training pairs. The rest videos (with different instances/background) serve as the test set. Then we warp the trained patch on each frame of the test videos with the fixed patch size ratio to evaluate the attack performance.

C. Ablation Study on MTD for SiamRPN++

We conduct and report in Table 6 the ablation study of the MTD loss on the SiamRPN++ tracker on the *person* category. For both of the dilation and shrinking attacks, we observe that MTD improves the attacking performance in three metrics. For instance, in the dilation attacks, the *success* metric improves by 11.3% by the incorporation of MTD; and this metric improves by 24.3% in the shrinking attack. Therefore, we can enhance the attack ability in the visual tracking attacks by additionally utilizing the MTD loss.

Metric	Dilation Attack (↓%)		Shrink Attack (↓%)	
	w/o MTD	w/ MTD	w/o MTD	w/ MTD
Success	31.2	**42.5**	14.5	**38.8**
Precision	24.7	**38.9**	14.0	**20.4**
Norm Precision	25.4	**36.0**	16.0	**17.8**

Table 6: Ablation study on the MTD loss on SiamRPN++ on the "person" category. The symbol "↓" denotes performance drop and larger values indicate stronger attacks.

To summarize, the proposed algorithm includes three losses: the MTD, Shape loss and TV. Without MTD, the attack performance degrades considerably. This is because MTD can effectively blind the Siamese-based trackers by de-matching the upstream representations. Without the Shape loss, it is difficult to control the output bounding box after attacks. This is because the Shape loss is to make attacks more controllable. Without the TV loss, the attack performance almost stays the same in digital simulations, but attacks will fail in physical experimental tests. This is because the TV loss is to smooth the patch texture to be physically feasible.

References

Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2018. Synthesizing robust adversarial examples. In *International conference on machine learning*, 284–293.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, 850–865. Springer.

Chen, S.-T.; Cornelius, C.; Martin, J.; and Chau, D. H. P. 2018. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 52–68. Springer.

Chen, X.; Yan, X.; Zheng, F.; Jiang, Y.; Xia, S.-T.; Zhao, Y.; and Ji, R. 2020. One-Shot Adversarial Attacks on Visual Tracking With Dual Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10176–10185.

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1625–1634.

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

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423.

Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; and Brendel, W. 2019. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations* .

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. *International Conference on Learning Representations* 1–11.

Guo, Q.; Xie, X.; Juefei-Xu, F.; Ma, L.; Li, Z.; Xue, W.; Feng, W.; and Liu, Y. 2019. SPARK: Spatial-aware Online Incremental Attack Against Visual Tracking. *arXiv preprint arXiv:1910.08681* .

Held, D.; Thrun, S.; and Savarese, S. 2016. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, 749–765. Springer.

Huang, L.; Gao, C.; Zhou, Y.; Xie, C.; Yuille, A. L.; Zou, C.; and Liu, N. 2020. Universal Physical Camouflage Attacks on Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 720–729.

Inkawhich, N.; Wen, W.; Li, H. H.; and Chen, Y. 2019. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7066–7074.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711. Springer.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* .

Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; and Yan, J. 2019. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4282–4291.

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

Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; and Ghanem, B. 2018. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 300–317.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, 8026–8037.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Sharif, M.; Bhagavatula, S.; Bauer, L.; and Reiter, M. K. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, 1528–1540.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. *International Conference on Learning Representations* 1–10.

Thys, S.; Van Ranst, W.; and Goedemé, T. 2019. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1–7.

Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; and Torr, P. H. 2019. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1328–1338.

Wiyatno, R. R.; and Xu, A. 2019. Physical adversarial textures that fool visual object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 4822–4831.

Wu, X.; Wang, X.; Zhou, X.; and Jian, S. 2019. STA: Adversarial Attacks on Siamese Trackers. *arXiv preprint arXiv:1909.03413* .

Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. 2017. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 1369–1378.

Yan, B.; Wang, D.; Lu, H.; and Yang, X. 2020a. Cooling-Shrinking Attack: Blinding the tracker with imperceptible noises. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 990–999.

Yan, X.; Chen, X.; Jiang, Y.; Xia, S.-T.; Zhao, Y.; and Zheng, F. 2020b. Hijacking Tracker: A Powerful Adversarial Attack on Visual Tracking. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2897–2901. IEEE.

Zhang, T.; and Zhu, Z. 2019. Interpreting adversarially trained convolutional neural networks. *International conference on machine learning* 1–10.

Zhao, Y.; Zhu, H.; Liang, R.; Shen, Q.; Zhang, S.; and Chen, K. 2019. Seeing isn't Believing: Towards More Robust Adversarial Attack Against Real World Object Detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 1989–2004.