International Journal of Computer Vision
https://doi.org/10.1007/s11263-024-02226-0

Robust Deep Object Tracking against Adversarial Attacks

Shuai Jia[1] · Chao Ma[1] · Yibing Song[2] · Xiaokang Yang[1] · Ming-Hsuan Yang[3]

Received: 5 September 2023 / Accepted: 16 August 2024

Abstract
Addressing the vulnerability of deep neural networks (DNNs) has attracted significant attention in recent years. While recent studies on adversarial attack and defense mainly reside in a single image, few efforts have been made to perform temporal attacks against video sequences. As the temporal consistency between frames is not considered, existing adversarial attack approaches designed for static images do not perform well for deep object tracking. In this work, we generate adversarial examples on top of video sequences to improve the tracking robustness against adversarial attacks under white-box and black-box settings. To this end, we consider motion signals when generating lightweight perturbations over the estimated tracking results frame-by-frame. For the white-box attack, we generate temporal perturbations via known trackers to degrade significantly the tracking performance. We transfer the generated perturbations into unknown targeted trackers for the black-box attack to achieve transferring attacks. Furthermore, we train universal adversarial perturbations and directly add them into all frames of videos, improving the attack effectiveness with minor computational costs. On the other hand, we sequentially learn to estimate and remove the perturbations from input sequences to restore the tracking performance. We apply the proposed adversarial attack and defense approaches to state-of-the-art tracking algorithms. Extensive evaluations on large-scale benchmark datasets, including OTB, VOT, UAV123, and LaSOT, demonstrate that our attack method degrades the tracking performance significantly with favorable transferability to other backbones and trackers. Notably, the proposed defense method restores the original tracking performance to some extent and achieves additional performance gains when not under adversarial attacks.

Keywords Visual tracking · Adversarial attack · Adversarial defense · Universal perturbations · Transferability

1 Introduction

The success of DNNs has significantly advanced object tracking in the last decade with numerous applications such as intelligent video surveillance, autonomous driving, robotic vision, and human-computer interaction. As object tracking can be posed as a sequential detection problem to distinguish the target and the background, existing deep object trackers often train DNN classifiers with positive and negative examples around the estimated target. Despite the demonstrated success, deep object trackers are typically vulnerable to the attack of adversarial examples. That is, adding imperceptible perturbations on input images can degrade the performance of the pretrained models seriously, as shown in image classification (Szegedy et al., 2014), object detection (Xie et al., 2017b), semantic segmentation (Xiao et al., 2018), and face recognition (Dong et al., 2019b). Given the vulnerability of DNNs, adversarial defense approaches (Sun et al., 2019; Xie et al., 2019) aim to improve the robustness against adversarial attacks. Nevertheless, existing studies on adversarial attack

Communicated by Ehsan Adeli.

✉ Chao Ma
chaoma@sjtu.edu.cn

Shuai Jia
jiashuai@sjtu.edu.cn

Yibing Song
yibingsong.cv@gmail.com

Xiaokang Yang
xkyang@sjtu.edu.cn

Ming-Hsuan Yang
mhyang@ucmerced.edu

[1] The MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China

[2] Alibaba DAMO Academy, Hangzhou, China

[3] University of California at Merced, Merced, CA, USA

Published online: 26 September 2024

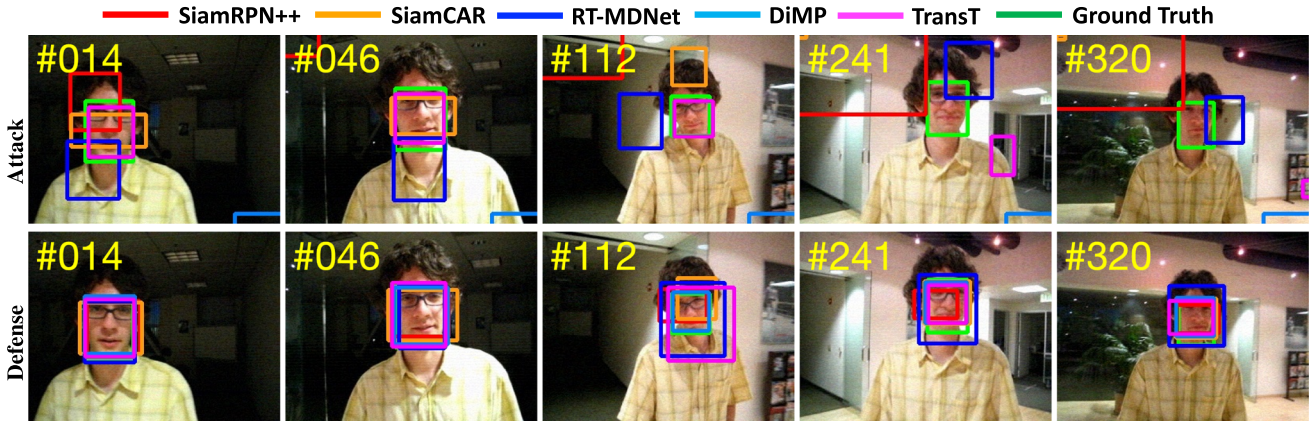


Fig. 1 Adversarial attack and defense for visual object tracking. On top of the five state-of-the-art deep trackers SiamRPN++ (Li et al., 2019), SiamCAR (Guo et al., 2020a), RT-MDNet (Jung et al., 2018), DiMP (Bhat et al., 2019), and TransT (Chen et al., 2021), we learn to generate adversarial examples to attack and defend them on the *David* sequence (Wu et al., 2015)

and defense mainly reside in static images, and considerably less attention has been paid to generating adversarial examples on top of video sequences for robust object tracking. The main challenges of exploiting adversarial attack and defense for robust visual tracking lie in two aspects. First, since visual trackers tend to sample candidates around the target, the limited search region increases the difficulty of attack. Recent attacks on multi-class classification are designed to fool one image, whereas attacks on object tracking require misclassifying multiple candidates simultaneously. In addition to the classifiers, deep trackers widely use a regression network to refine bounding boxes. Tracking performance would significantly degrade when adversarial attacks are successfully applied to the regression module. Existing works investigate the regression attack for various tasks, including object detection (Gupta et al., 2021; Lu et al., 2017; Wang et al., 2021), single object tracking (Guo et al., 2020b, 2021; Yan et al., 2020) and multiple object tracking (Jia et al., 2019; Zhou et al., 2023) (Lin et al., 2021). However, directly transferring these methods yields limited attack performance since existing trackers locate the search regions around the target. Second, motion consistency between frames causes existing attacks on static images to perform poorly, where a tracker can relocate the target after a few frames. Temporal consistency is rarely investigated to improve attack success rates against video sequences.

In this work, we improve the robustness of state-of-the-art deep trackers against adversarial attacks by involving both the spatial and temporal domains. Specifically, we do not modify existing deep trackers and keep main components such as sampling schemes unchanged. Existing adversarial attacks can be categorized as white-box or black-box based on whether the attackers know the internal architectures of the attacked models. For white-box attacks, we learn perturbations and inject them into input frames, yielding indistinguishable binary samples (i.e., some samples are incorrect). We use these binary adversarial examples to retrain classifiers to degrade their performance. Specifically, we minimize the classification loss difference between the correct and incorrect binary samples. Meanwhile, we randomly shift and rescale ground truth boxes to attack the regression network. On the other hand, when considering the temporal consistency between frames, we use the learned perturbations in the current frame to initialize the perturbation learning in the next frame. Applying the temporally generated perturbations to every frame further degrades the performance of deep trackers. Figure 1 shows one example that the state-of-the-art deep trackers under adversarial attacks drift rapidly (see the first row). For black-box attacks, we generate the transferable perturbations with the victim tracker and transfer them into various backbones and targeted trackers frame-by-frame. To strengthen its transferability, we retain the distribution of adversarial perturbations from previous frames and constantly fuse them into more transferable and robust perturbations. The combination of perturbations from different frames maintains the diversity of the adversary and benefits its transferability.

Although the proposed attack method significantly degrades the performance of various trackers, the attack performance of generated perturbations relies heavily upon the context of each frame, leading to a high computational cost and a low attack speed. Similar to Moosavi-Dezfooli et al. (2017), we further propose to train universal adversarial perturbations (UAP) for visual object tracking, where we only add the same perturbations into each frame of video sequences. The attack speed is greatly improved since the whole process of generating UAP is trained offline with training datasets. Only a simple addition operation is involved for

each frame, bringing the speed close to the original tracking speed.

We further explore adversarial defense to improve the robustness of deep trackers against adversarial attacks. Note that the adversarial perturbations are assumed to be unknown. We aim to estimate the unknown perturbations in the input videos and learn to eliminate their effects during tracking. The estimation process is similar to the attack process, but the involved samples differ. As an example shown in Fig. 1, we perform the proposed adversarial attack and defense approaches on five state-of-the-art deep tracking methods (Li et al., 2019; Guo et al., 2020a; Jung et al., 2018; Bhat et al., 2019; Chen et al., 2021). Besides, the proposed defense approach achieves additional performance gains when the trackers are not under adversarial attacks. Our defense module can estimate the naturally occurring adversarial perturbations during input images, such as noise in the imaging process.

Early findings of this work are presented in Jia et al. (2020, 2021) and the main differences between this manuscript and our conference paper are:

- In addition to the original white-box attack (Jia et al., 2020), we propose a transferring attack approach without access to the targeted tracker, i.e., a black-box attack. Building on the original attack, our transferring attack generates dense adversarial perturbations in the spatiotemporal domain and transfers them into unknown trackers, resulting in strong attack transferability across various backbones and trackers.
- Previous gradient-based attack (Jia et al., 2020) relies heavily on image content, and the decision-based attack (Jia et al., 2021) needs to query the tracker to get the feedback constantly, leading to heavy computational loads and slow attack speeds. To address this, we exploit universal adversarial perturbations and directly apply them to all frames, thereby accelerating the attack speed and almost maintaining the original tracking speed.
- Extensive experiments with diverse architectures of trackers (i.e., SiamRPN++ (Li et al., 2019), SiamCAR (Guo et al., 2020a), RT-MDNet (Jung et al., 2018), DiMP (Bhat et al., 2019), and TransT (Chen et al., 2021)) on the six benchmark datasets demonstrate that our attack causes considerable drops on various trackers. Furthermore, our defense can exclude the adversarial perturbations to restore the tracking performance.

2 Related Work

In this section, we first introduce tracking methods closely related to this work. We then review recent adversarial attack methods, especially for visual object tracking. In addition, we discuss the recent adversarial defense models.

2.1 Deep Object Tracking

Deep object tracking can be roughly categorized as one-stage regression-based methods and two-stage detection-based methods. The regression-based methods typically learn correlation filters over CNN features to locate target objects as in Ma et al. (2015). Numerous methods (Wang et al., 2015; Held et al., 2016; Valmadre et al., 2017; Song et al., 2017; Lu et al., 2018; Wang et al., 2019, 2020; Shen et al., 2022; Yan et al., 2022; Ma et al., 2022; Borsuk et al., 2022) have since been proposed to improve tracking performance in different aspects, including feature hedging (Qi et al., 2016), continuous convolution (Danelljan et al., 2016), particle filter integration (Zhang et al., 2017), efficient convolution (Danelljan et al., 2017), spatiotemporal regularization (Li et al., 2018b), RoI pooling (Sun et al., 2019) and correlation-aware (Xie et al., 2022). On the other hand, two-stage tracking-by-detection approaches first generate multiple candidate regions and then classify each as either the target or the background. Siamese-based methods (Zhang & Peng, 2019b; Chen et al., 2022; Lai et al., 2023; Zhang et al., 2020) are one of the widely used trackers, which generally consist of classification and regression branches. Some works (Li et al., 2018a; Zhu et al., 2018; Li et al., 2019) utilize the region proposal network (RPN) (Ren et al., 2015) to conduct feature fusion with depthwise correlation, yielding more accurate tracking results. Other works (Guo et al., 2020a; Chen et al., 2020b) exploit anchor-free architectures to regress the bounding boxes, avoiding hyper-parameter tuning. In addition, recent methods (Song et al., 2022; Gao et al., 2022; Wei et al., 2023; Gao et al., 2023; Lai et al., 2023) use Transformer (Vaswani et al., 2017) or Vision Transformer (ViT) (Dosovitskiy et al., 2020) to learn the spatial and temporal features for object tracking, including CNN-Transformer based trackers (Chen et al., 2021; Wang et al., 2021a; Yan et al., 2021; Yu et al., 2021; Cao et al., 2021; Mayer et al., 2022; Xing et al., 2022; Ma et al., 2022) and fully-Transformer based trackers (Xie et al., 2021; Lin et al., 2021; Cui et al., 2022; Chen et al., 2022). From the perspective of model updates, existing deep tracking approaches can be classified as either offline or online. Offline trackers (Bertinetto et al., 2016; Li et al., 2018a; Zhu et al., 2018; Guo et al., 2020a; Chen et al., 2020b) do not update model parameters during the inference stage, leading to a higher tracking speed. In contrast, online trackers (Han et al., 2017; Danelljan et al., 2019; Bhat et al., 2019; Song et al., 2018; Dai et al., 2020; Zhang et al., 2019a; Pu et al., 2018) constantly update model parameters by learning the CNN features from previous frames. The MDNet approaches (Nam & Han, 2016; Jung et al., 2018) consider tracking as a classifi-

cation task to distinguish the target and background. During inference, they incrementally collect positive and negative samples to enhance the discriminative ability of the classifier. In this work, we evaluate our attack and defense approaches on five our representative state-of-the-art trackers, including one Siamese-based tracker (Li et al., 2019) without online updates, one anchor-free tracker (Guo et al., 2020a), one classification-based tracker (Jung et al., 2018) with online updates, one discriminative-based tracker (Bhat et al., 2019) and one transformer-based tracker (Chen et al., 2021). We aim to demonstrate the general effectiveness of adversarial attack and defense on diverse architecture trackers.

2.2 Adversarial Attack

Recent studies (Goodfellow et al., 2015; Szegedy et al., 2014) demonstrate that deep networks are vulnerable to adversarial examples. Despite state-of-the-art performance on natural input images, the pre-trained networks perform poorly given intentionally generated adversarial examples. Existing adversarial attack methods mainly fall into white-box and black-box attacks. The deep models are assumed to be known in white-box attacks (Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016), whereas they are unknown in black-box attacks (Ilyas et al., 2018; Liu et al., 2017). Specifically, black-box attacks are categorized into the following types, including transferable attacks (Sun et al., 2023; Dong et al., 2019a), model stealing (Sun et al., 2022; Zhou et al., 2020; Wang et al., 2021b), gradient estimation (Brendel et al., 2018; Cheng et al., 2018), etc. In addition to algorithmic attacks, physical attack methods generate real-world objects to lead models to misclassification. These are typically useful to examine the robustness of automotive driving in road sign scenarios (Kurakin et al., 2017; Eykholt et al., 2018; Wiyatno & Xu, 2019; Ding et al., 2021). In object tracking, some recent works (Yan et al., 2020; Jia et al., 2020; Chen et al., 2020a; Liang et al., 2020; Guo et al., 2020a, 2021; Jia et al., 2021) have explored the vulnerability of deep networks. Yan et al. (2020) propose a cooling-shrinking loss to generate imperceptible noises, which cool hot regions on the heatmaps and shrink the predicted bounding box. Chen et al. (2020a) present a one-shot adversarial attack method by only adding the perturbations into the template branch. It optimizes the confidence and feature loss and leverages the dual attention mechanisms. SPARK (Guo et al., 2020b) utilizes past fewer frames to generate an incremental perturbation to achieve targeted attacks. On the other hand, FAN (Liang et al., 2020) trains an end-to-end network to integrate the drift loss and the feature loss to attack the Siamese-based tracker, and (Wiyatno & Xu, 2019) propose a physical adversarial attack to generate adversarial textures to drift the targeted tracker. ABA (Guo et al., 2021) formulates the adversarial attack with the motion blur pattern, causing a significant drop in tracking accuracy. In this work, we consider multiple frames and propose a temporal motion attack. Our method performs well on white-box attacks and yields favorable attack transferability to other backbones, targeted trackers, and black-box attacks.

2.3 Adversarial Defense

Defending DNNs against adversarial attacks can be regarded as robustly learning DNNs with adversarial examples. Existing adversarial defense methods can be mainly categorized into two classes. The first class of methods is based on adversarial training. Tramèr et al. (2018) propose adversarial training by adding the adversarial examples into the original training dataset to retrain the model. This defense method is effective against the adversarial attack that generates the adversarial examples. Madry et al. (2018) consider adversarial training as a min-max optimization to train the model solely with adversarial examples. Introducing adversarial examples into the training process generally causes a performance drop. This is typically a trade-off between one model's performance and robustness. In (Xie et al., 2019) Xie et al. propose a feature denoising module to enhance the model robustness and even improve its original performance slightly. Numerous attempts have been made to purify adversarial examples to defend against the attack. From this perspective, adversarial examples produce noise on features to distract the network inference process. In Liao et al. (2018), denoising algorithms are proposed to eliminate the effect of noise. In addition, images are transformed to be non-differentiable in Guo et al. (2018) to resist adversarial attacks. Xie et al. (2017a) use random resizing and padding during the inference time to mitigate adversarial effects. Unlike existing attack and defense methods, we attack deep trackers' classification and regression modules to decrease accuracy. Then, we gradually estimate adversarial perturbations and eliminate their effect on input images without modifying existing deep trackers.

3 Proposed Algorithms

In this section, we present how to perform adversarial attacks and defense for visual tracking. Adversarial attacks include gradient-based white-box attacks, transfer-based black-box attacks, and universal attacks. Given an input video sequence and a labeled bounding box in the initial frame, we generate adversarial examples spatiotemporally to decrease tracking accuracy. Meanwhile, our defense algorithm learns to estimate unknown adversarial perturbations and eliminate their effect from input sequences. Figure 2 illustrates the overall variation of adversarial perturbations during adversarial attack and defense.

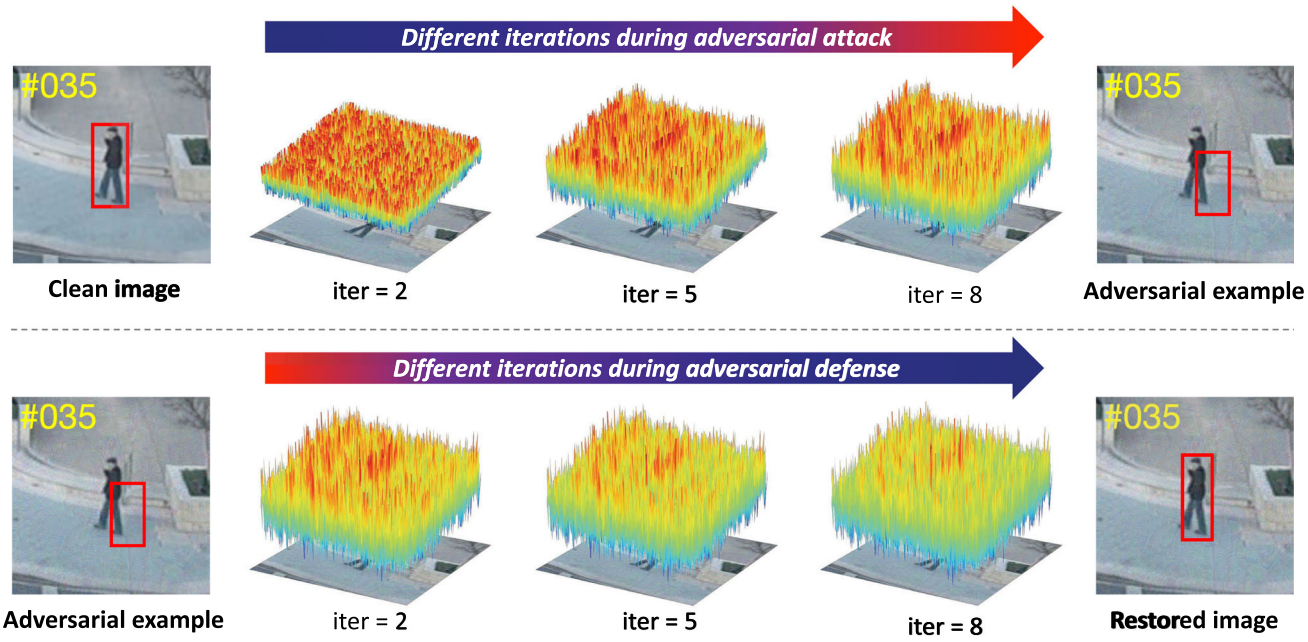


Fig. 2 Variation of adversarial perturbations during attack and defense. The 3D response map above the image represents the difference between the clean image and the adversarial example at the current iteration. In adversarial attacks, the perturbations increase along with training iterations. In adversarial defense, the perturbations decrease when training iteration increases

3.1 Generating Adversarial Examples

We generate adversarial perturbations based on deep trackers' input frame and output response, i.e., classification scores or regression maps. These perturbations are then added to the input frame to generate adversarial examples. Deep trackers usually employ a DNN architecture containing two branches in the tracking-by-detection framework. In the first branch, the sampled candidate regions are classified as either the target or background, while another branch is regressed for precise localization. We denote an input frame by I, a candidate number by N, a binary classification loss by L_c, a bounding box regression loss by L_r, and correct classification label and regression label by p_c and p_r, respectively. Both labels p_c and p_r are predicted by the tracking result S^{t-1} from the last frame, while S^1 is the ground-truth annotation in the initial frame. The original loss function of the tracking-by-detection network is:

$$\mathcal{L}(I, N, \theta) = \sum_{n=1}^{N} [L_c(I_n, p_c, \theta) + \lambda \cdot L_r(I_n, p_r, \theta)], \quad (1)$$

where I_n is one candidate region in the image, λ is a fixed weight parameter, and θ denotes the network parameters to be optimized during training.

When generating adversarial perturbations, we expect the networks to make inaccurate inferences. We create a pseudo classification label $p_c^\star$ and a pseudo regression label $p_r^\star$. The adversarial loss is set to make L_c and L_r the same when we use correct and pseudo labels. The adversarial loss can be formulated as:

$$\begin{aligned}\mathcal{L}_{adv}(I, N, \theta) = \sum_{n=1}^{N} \{ & [L_c(I_n, p_c, \theta) - L_c(I_n, p_c^\star, \theta)] \\ & + \lambda \cdot [L_r(I_n, p_r, \theta) - L_r(I_n, p_r^\star, \theta)]\},\end{aligned} \quad (2)$$

where θ is fixed because the network is in the inference stage. The adversarial loss $\mathcal{L}_{adv}$ reflects the loss similarity between using correct and pseudo labels. When minimizing $\mathcal{L}_{adv}$, the predictions will be close to pseudo labels, and the performance will degrade rapidly.

We set pseudo labels specifically for each branch. In $p_c^\star$, two elements (i.e., 0 and 1) indicate the probabilities of the input belonging to the target and background. We set $p_c^\star$ by reversing the elements of p_c to confuse the classification branch. On the other hand, p_r consists of four elements (x_r, y_r, w_r, h_r) representing the target location. We set $p_r^\star$ by adding a random distance offset and a random scale variation to p_r. Each element of $p_r^\star$ can be written as:

$$\begin{aligned} x_r^\star &= x_r + \delta_{\text{offset}} \\ y_r^\star &= y_r + \delta_{\text{offset}} \\ w_r^\star &= w_r * \delta_{\text{scale}} \\ h_r^\star &= h_r * \delta_{\text{scale}}, \end{aligned} \quad (3)$$

where δ_{offset} and δ_{scale} indicate the random distance offset and random scale variation, respectively.

After computing the adversarial loss using Eq. 2, we take partial derivatives of the adversarial loss concerning input I. Formally, the partial derivative R is computed as:

$$R = \frac{\partial \mathcal{L}_{adv}}{\partial I}. \tag{4}$$

To reduce outlier effects, we pass R into a sign function. Given an input frame I, we use M iterations to generate the final adversarial perturbations. The output of the last iteration is added into the input frame, which can be written as follows:

$$I_{m+1} = I_m + \alpha \cdot sign(R_m), \tag{5}$$

where $\alpha = \frac{\epsilon}{M}$ is a weight parameter, ϵ is the maximum value of the perturbations, m indicates the iteration index, I_m is the input frame for the m-th iteration, M denotes the final iteration number, and $\alpha \cdot sign(R_m)$ is the perturbations generated during the m-th iteration. After M iterations, the final adversarial example is I_M.

As video frames are temporally coherent, we consider the adversarial attack in the spatiotemporal domain. When an input video sequence has T frames, we use the learned perturbations in the last frame as initialization for the current frame. Specifically, for the t-th frame, we use perturbations from the last frame to initialize I^t, which can be written as:

$$I_1^t = I_1^t + (I_M^{t-1} - I_1^{t-1}), \tag{6}$$

where $I_M^{t-1} - I_1^{t-1}$ is the perturbation from the last frame. We gradually update I^t by using Eq. 5 to generate the final perturbations for the t-th frame. Algorithm 1 summarizes the main steps for generating adversarial examples. Note that we use the IoU metric (Ren et al., 2016) to assign labels for candidates.

Algorithm 1: Adversarial Example Generation

Input: input video V with T frames;
target location S^1;
Output: adversarial examples of T frames;
for $t = 2$ **to** T **do**
  Get current frame I_1^t;
  if $t \neq 2$ **then**
    Update I_1^t via Eq. 6;
  end
  for $m = 1$ **to** M **do**
    Create p_c and p_r via IoU ratios between candidates and target location S^{t-1};
    Create $p_c^\star$ by reversing elements of p_c;
    Create $p_r^\star$ via Eq. 3;
    Generate adversarial loss via Eq. 2;
    Update I_m^t via Eq. 5;
  end
  return I_M^t;
end

3.2 Transferring Attacks

In contrast to white-box attacks, black-box attacks have no access to the architecture and parameters of the victim model. For visual object tracking, IoU Attack (Jia et al., 2021) explores the decision-based black-box attacks to iteratively increase the noise according to the feedback of the targeted models. However, it requires consistently querying the victim model to calculate the IoU scores and adjust the directions and magnitudes of noise, which leads to heavy and time-consuming computations. In this paper, we adopt transfer-based attacks as an effective black-box attack method without the feedback of target models. Specifically, we implement our attack method on the victim tracker to generate the adversarial perturbations and transfer them to the targeted tracker frame-by-frame. When generating the transferable adversarial perturbations, we remove the original sign function of gradients in Eq. 5 and calculate the continuous perturbations in the direction of loss gradients as:

$$G = \nabla \mathcal{L}_{adv} \cdot \frac{\mathcal{L}_{adv}}{||\nabla \mathcal{L}_{adv}||_2^2}, \tag{7}$$

where ∇ is the partial derivative of input I. Instead of using the sign function that destroys the internal distributions of loss gradient on different pixels, we compute the ℓ_2 norm to regularize the loss gradient and maintain its weights. In addition, the temporal consistency between frames also benefits the attack transferability of perturbations for tracking. Thus, we fuse the perturbations from previous frames as follows:

$$\hat{G}^t = G^t + \delta \cdot (G^{t-1} + \delta \cdot (G^{t-2} + \delta \cdot (G^{t-3} + \cdots \delta \cdot G^0))), \tag{8}$$

where t is the frame index, and δ is a hyper-parameter to adjust the weight of adversary from various frames temporally. We then transfer the generated adversarial perturbations frame-by-frame and use $\text{Trunc}(I^t + \hat{G}^t)$ to truncate every pixel value by [0, 255].

3.3 Universal Adversarial Perturbation

The above-described attack belongs to image-dependent methods that generate individual perturbations for each frame. Although the attack significantly degrades tracking performance, it relies heavily on image content. The attack process has a high computational load, resulting in weak transferability and slow attack speed. Motivated by UAP (Moosavi-Dezfooli et al., 2017) for image classification and IoU attack (Jia et al., 2021), we present an efficient attack method that carries out universal adversarial perturba-

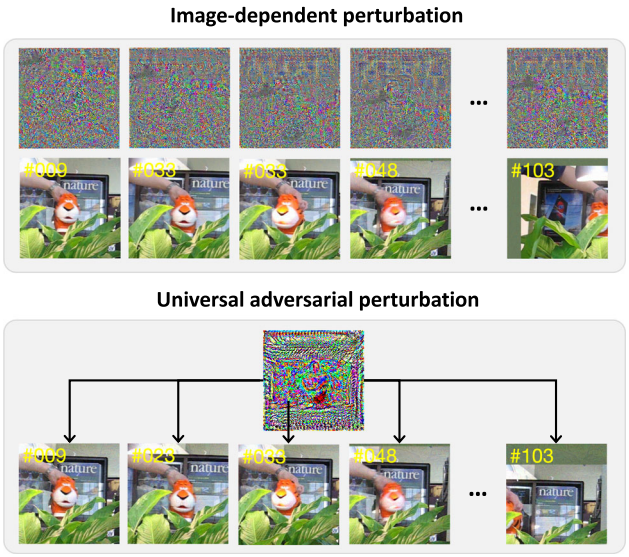


Fig. 3 The comparison of image-dependent perturbation and universal adversarial perturbation

tions (UAP), which can be applied to all frames in tracking videos. Different from UAP (Moosavi-Dezfooli et al., 2017), which focuses on identifying the boundaries of different classes and finding a position that misclassifies more images, our method is designed for visual tracking by employing our adversarial attack losses that involve both classification and regression branches simultaneously. The crafted UAP is generated offline by continuously querying the training dataset. It can obtain independent data distribution and is suitable to apply to any video frame. Furthermore, the attack speed maintains its original tracking speed since only a few additional operators are involved. The differences between image-dependent perturbations and universal adversarial perturbations are illustrated in Fig. 3.

During the training process of UAP, the adversarial loss is consistent with Eq. 2. When computing the perturbation gradient, we maintain the distribution of gradients from each training pair following the transferring attack in Eq. 7. First, we initialize the UAP with the first training pair. In the following pair, we update the perturbations of UAP on the current pair. Specifically, we compute the IoU score between the current result S^m and grounding truth label B^m. If the IoU score is less than a predefined threshold, it implies that the UAP has already drifted the tracker successfully in the current frame, and we skip this frame. Otherwise, we update the UAP denoted by adding the generated perturbations from current frames as follows:

$$\mathcal{U}^m = \mathcal{U}^{m-1} + \zeta \cdot G, \tag{9}$$

where $\mathcal{U}$ and G denote the perturbations of the UAP and the current frame, respectively, and ζ is a weight parameter.

Algorithm 2: Universal Adversarial Perturbation

Input: input M training pairs;
grounding truth boxes B^m;
initialized perturbation $\mathcal{U}^0$
Output: universal adversarial perturbation (UAP);
for $m = 1$ **to** M **do**
  Add the UAP $\mathcal{U}^{m-1}$ on current frame I^m;
  Get the tracking result S^m;
  Calculate IoU score between S^m and B^m;
  if $S_{\text{IoU}} \leq S_{\text{th}}$ **then**
    Generate adversarial loss via Eq. 2;
    Calculate the adversarial perturbations via Eq. 7;
    Update the UAP $\mathcal{U}^m$ via Eq. 9;
  else
    Skip this frame;
  end
end
return $\mathcal{U}^M$;

We repeat the above process and obtain final outputs $\mathcal{U}^M$. The whole process of generating the universal adversarial perturbation is summarized in Algorithm 2.

3.4 Adversarial Defense

We propose an adversarial defense method against adversarial attacks for object tracking. The motivation to disrupt the distribution of adversarial perturbations is intuitive. From Eq. 4, we observe that perturbations originate from partial derivatives. Instead of adding perturbations to the input frame to decrease tracking accuracy, we gradually estimate the potential unknown perturbations and subtract them from the input frame. Although the gradients from defense are inconsistent with those from attack, they are still effective in mitigating the impact of adversarial perturbations. As a result, the effect of unknown perturbations will be eliminated, helping DNNs make correct inferences and restoring tracking performance. Similar to our attack, we defend adversarial examples without updating the networks.

Given an input frame I with unknown adversarial perturbations, we generate correct and pseudo labels according to the predicted location S^{t-1} from the previous frame. The label generation process is similar to that in Sect. 3.1 except that the candidates during defense are resampled based on the adversarial examples. We then estimate the adversarial loss using Eq. 2 and compute the partial derivatives via Eq. 4. We apply partial derivatives R on the input frame I via the following operation:

$$I_{m+1} = I_m - \text{Trunc}_{\beta \cdot R \in [-\hat{\alpha}, \hat{\alpha}]}(\beta \cdot R), \tag{10}$$

where β is a weight parameter, $\text{Trunc}(\cdot)$ is a truncation function to constrain the values of $\beta \cdot R$ within the range between $-\hat{\alpha}$ and $\hat{\alpha}$. The parameter $\hat{\alpha}$ resembles to the parameter α

Algorithm 3: Adversarial Example Defense

Input: input video V with T adversarial examples; target location S^1;
Output: adversarial examples of T frames;
for $t = 2$ **to** T **do**
  Get current frame I_1^t;
  if $t \neq 2$ **then**
    Update I_1^t via Eq. 11;
  end
  for $m = 1$ **to** M **do**
    Create p_c and p_r via IoU ratios between candidates and target location S^{t-1};
    Create $p_c^\star$ by reversing elements of p_c;
    Create $p_r^\star$ via Eq. 3;
    Generate adversarial loss via Eq. 2;
    Update I_m^t via Eq. 10;
  end
  return I_M^t;
end

in Eq. 5. Since the perturbation is unknown during defense, we empirically set different values for these two parameters for various attacks. When the input videos contain T frames, we transfer the perturbations from the last frame to the current frame as initialization. For the t-th frame, we update it initially as:

$$I_1^t = I_1^t - \gamma \cdot (I_1^{t-1} - I_M^{t-1}), \tag{11}$$

where γ is a weight parameter. The pseudo code of adversarial defense is shown in Algorithm 3.

Visualizations. We utilize the SiamRPN++ (Li et al., 2019) to visualize how adversarial perturbations vary during different iterations in Fig. 2. Given an input frame, we visualize the adversarial perturbations during attacks. Along with the training iterations, the variation of perturbations increases as well. The adversarial examples lead SiamRPN++ to drift rapidly. When defending this adversarial example, we observe that the variation of the perturbations decreases when training iteration increases. It indicates that the proposed defense method effectively estimates and excludes the perturbations, which helps alleviate performance drops caused by adversarial attacks.

4 Experimental Results

In this section, we first present the implementation details of the proposed method and introduce the deployment of deep trackers during experiments. Then, we evaluate the attack and defense methods on benchmark datasets. We measure the attack transferability among different backbones and trackers as black-box attacks. To improve the attack speed, we train universal adversarial perturbations and add them to all frames to evaluate the attack performance. In addition, we conduct ablation studies on the variants of our attack method. Finally, we compare our attack and defense with existing methods.

4.1 Implementation Details

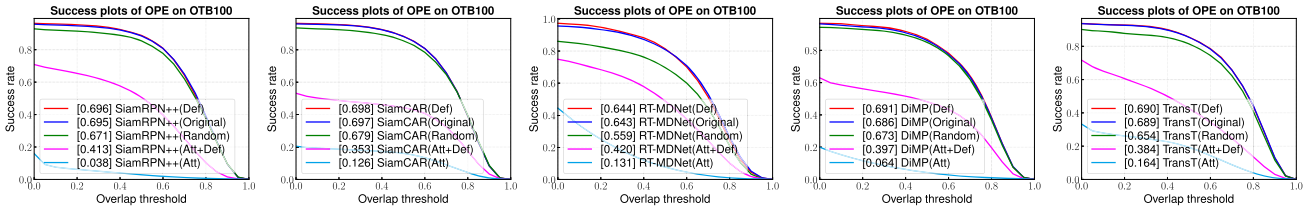
We apply our methods to five representative deep trackers covering a variety of tracking architectures: SiamRPN++ (Li et al., 2019), SiamCAR (Guo et al., 2020a), RT-MDNet (Jung et al., 2018), DiMP (Bhat et al., 2019), and TransT (Chen et al., 2021). The maximum variation value of each pixel in the perturbations is set to 10 (i.e., $\epsilon = 10$) for attacks and set to 5 (i.e., $\epsilon = 5$) for defense. The perturbations are quantized to integers for attack and defense to preserve the image quality. When computing IoU ratios between candidates and the target location S_{t-1}, we follow the threshold setting of trackers to distinguish the positive and negative samples. Note that deep trackers' parameters remain fixed during adversarial attacks and defense. All experiments are performed on a PC with an Intel i9 3.6GHz CPU and an NVIDIA RTX 2080Ti GPU. The source codes of the proposed methods are available at https://vision.sjtu.edu.cn/rtaa/rtaa.html.

4.2 Deployment of Deep Trackers

To illustrate the generality of our methods, we apply the proposed adversarial attack and defense to five state-of-the-art trackers (Li et al., 2019; Guo et al., 2020a; Jung et al., 2018; Bhat et al., 2019; Chen et al., 2021). From the perspective of online updates, we consider RT-MDNet (Jung et al., 2018) and DiMP (Bhat et al., 2019) as representative online trackers, while other trackers utilize offline pretrained models. We select SiamRPN++ (Li et al., 2019) and SiamCAR (Guo et al., 2020a) to represent the anchor-based and anchor-free trackers. In addition, we adopt DiMP (Bhat et al., 2019) as an instance of discriminative-based tracker and TransT (Chen et al., 2021) as an illustration of transformer trackers.

SiamRPN++. There are two output branches in SiamRPN++ for classifying and regressing proposals. During tracking, SiamRPN++ does not perform online updates. We use Algorithm 1 and Algorithm 3 to generate and defend adversarial examples when processing each frame. As the inputs contain a template and search patch, we take partial derivatives concerning the search patch when computing Eq. 4 for both attack and defense.

SiamCAR. The SiamCAR framework consists of one Siamese subnetwork for feature extraction and one classification-regression subnetwork for bounding box prediction. Different from SiamRPN++, SiamCAR is an anchor-free tracker without region proposal networks. We use the adversarial loss's original setting during training to adopt the cross-entropy loss for classification and the IoU loss for regression in Eq. 4. When processing each frame, we use Algorithm 1 and Algorithm 3 to perform adversarial attack and defense.



(a) SiamRPN++ (b) SiamCAR (c) RT-MDNet (d) DiMP (e) TransT

Fig. 4 Tracking performance of adversarial attack and defense methods on the OTB100 dataset (Wu et al., 2015). 'Att' and 'Def' denote the adversarial attack and defense, respectively, and 'Random' denotes random perturbations

RT-MDNet. The CNN module in RT-MDNet is only for classification. RT-MDNet online updates its model during tracking by collecting samples from previous frames. We generate adversarial examples and perform prediction and model updates. This configuration aims to analyze whether online updates effectively defend adversarial examples. We generate and defend adversarial examples using Algorithm 1 and Algorithm 3, except that we remove the regression terms when computing Eq. 2.

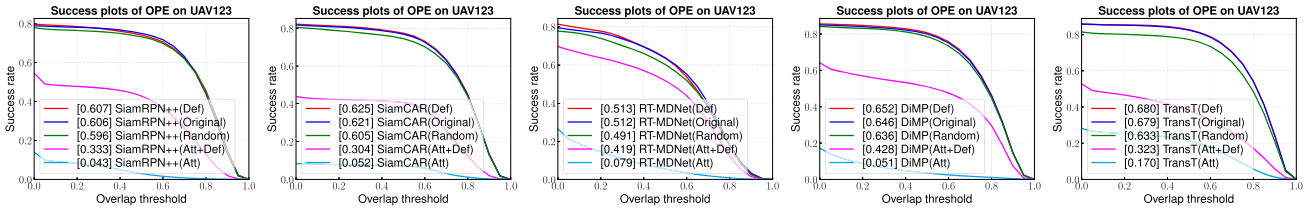
DiMP. Similar to the pipeline of Discriminative Correlation Filter (DCF), DiMP proposes an end-to-end online update tracking architecture. It enhances the discriminative capability to handle target and background appearance information with CNN features. For adversarial loss, we follow its original discriminative learning loss. Similarly, we employ Algorithm 1 and Algorithm 3 to implement the adversarial attack and defense.

TransT. Inspired by Transformer (Vaswani et al., 2017), TransT presents an attention-based feature fusion network with the architecture of Transformer. TransT comprises the designed attention-based fusion mechanism and the classification and regression branches. For adversarial loss, we adopt its original setting, which uses binary cross-entropy loss for classification and combines the generalized IoU loss (Rezatofighi et al., 2019) and ℓ_1 norm loss for regression. The processes of attack and defense follow Algorithm 1 and Algorithm 3.

4.3 Benchmark Datasets

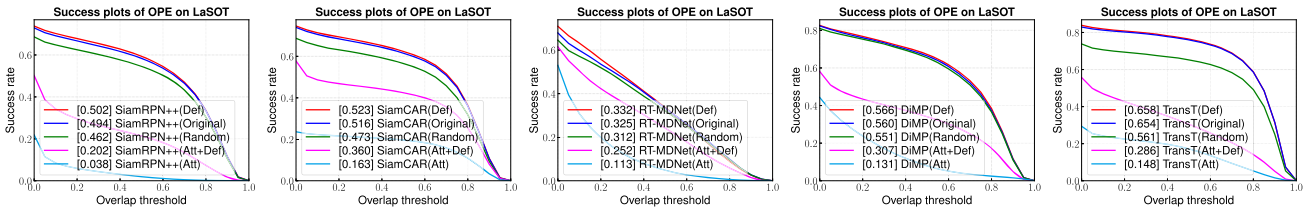
OTB100. There are 100 video sequences in the OTB100 (Wu et al., 2015) dataset with substantial target variations. We use the one-pass evaluation (OPE) with success plots for experiments. We first report the tracking results on the original dataset to evaluate our adversarial attack and defense methods. Next, we attack various trackers by adding adversarial perturbations to input video sequences. Meanwhile, we compare the attack performance by adding random perturbations containing the same variations as those of adversarial perturbations. Figure 4 illustrates that our attack method reduces the AUC scores of SiamRPN++ from 0.695 to 0.038 (i.e., a 94.5% decrease); SiamCAR from 0.697 to 0.126 (i.e., an 81.9% decrease); RT-MDNet from 0.643 to 0.131 (i.e., a 79.6% decrease); DiMP from 0.686 to 0.064 (i.e., a 90.7% decrease); and TransT from 0.689 to 0.164 (i.e., a 76.2% decrease). The trackers under adversarial attacks perform significantly worse than those under random perturbations. This suggests the effectiveness of our adversarial attack method on various types of targeted trackers. On the other hand, we implement the proposed defense method to subtract adversarial perturbations and restore the tracking performance. The AUC scores in success plots are restored to 0.413 for SiamRPN++, 0.353 for SiamCAR, 0.420 for RT-MDNet, 0.397 for DiMP, and 0.384 for TransT, respectively. This demonstrates that our defense method can effectively remove the perturbations to restore tracking performance. In addition, we apply our defense method to original tracking sequences and slightly improve the AUC scores for all trackers. These results show that perturbations exist in real-world scenarios during the image formation process (e.g., camera sensor noise and transformation from optical perception to digital storage). The proposed defense method effectively estimates these naturally existing perturbations and eliminates their effects.

UAV123. The UAV123 (Mueller et al., 2016) dataset contains 123 sequences with more than 110K frames, which are captured from low-altitude unmanned aerial vehicles. We adopt the success and precision plots to evaluate the performance. Figure 5 illustrates the success plots of these five trackers. Under the adversarial attacks, the AUC scores drop dramatically from 0.606 to 0.043 (i.e., a 92.9% decrease) for SiamRPN++; from 0.621 to 0.052 (i.e., a 91.6% decrease) for SiamCAR; from 0.512 to 0.079 (i.e., an 84.6% decrease) for RT-MDNet; from 0.646 to 0.051 (i.e., a 92.1% decrease) for DiMP; and from 0.679 to 0.170 (i.e., a 75.0% decrease) for TransT. For adversarial defense, the AUC scores are restored to 55.0% for SiamRPN++; 49.0% for SiamCAR; 81.8% for RT-MDNet; 66.3% for DiMP; and 47.6% for TransT of their original performance since the target objects on UAV123 mostly undergo significant shape changes. These results demonstrate the effectiveness of the proposed attack and defense method.



(a) SiamRPN++ (b) SiamCAR (c) RT-MDNet (d) DiMP (e) TransT

Fig. 5 Tracking performance of adversarial attack and defense methods on the UAV123 dataset (Mueller et al., 2016). 'Att' and 'Def' denote the adversarial attack and defense, respectively, and 'Random' denotes random perturbations



(a) SiamRPN++ (b) SiamCAR (c) RT-MDNet (d) DiMP (e) TransT

Fig. 6 Tracking performance of adversarial attack and defense methods on the LaSOT dataset (Fan et al., 2019). 'Att' and 'Def' denote the adversarial attack and defense, respectively, and 'Random' denotes random perturbations

LaSOT. LaSOT (Fan et al., 2019) is a large dataset with 1400 videos. The average length of video sequences is more than 2,500 frames, which is longer than most datasets and more challenging. We adopt the success rate as our evaluation metric. Figure 6 illustrates the tracking performance of our adversarial attack and defense methods on LaSOT. The tracking performance drops dramatically after attacks for both trackers, where the success rates are reduced from 0.494 to 0.038 (i.e., a 92.3% decrease) for SiamRPN++; from 0.516 to 0.163 (i.e., a 68.4% decrease) for SiamCAR; from 0.325 to 0.113 (i.e., a 65.2% decrease) for RT-MDNet; from 0.560 to 0.131 (i.e., a 76.6% decrease) for DiMP; and from 0.654 to 0.148 (i.e., a 77.4% decrease) for TransT. After applying our defense method to the adversarial examples, the AUC scores are improved to 81.2% for SiamRPN++; 54.7% for SiamCAR; 55.2% for RT-MDNet; 54.8% for DiMP; and 48.3% for TransT of their original performance. Our defense method helps these trackers counter adversarial attacks. In addition, our defense method slightly improves the tracking performance when deploying on the original sequences.

VOT datasets. There are all 60 challenging video sequences in the VOT2019 (Kristan et al., 2019), VOT2018(Kristan et al., 2018) and VOT2016 (Kristan et al., 2016) datasets. The VOT toolkit reinitializes the tracker if it loses the target object during five consecutive frames. The evaluation metrics of VOT are accuracy, robustness, and expected average overlap (EAO). The accuracy represents the average overlap ratio, and the number of reinitializations measures robustness. EAO measures trackers' overall performance. Table 1 illustrates our attack and defense methods for SiamRPN++, SiamCAR, RT-MDNet, DiMP, and TransT for three VOT datasets. Our attack method reduces the EAO scores by about 60 % for all five trackers on three VOT datasets. When applying our defense method, trackers' performance improves to about 50% of their original EAO scores. For VOT2019, our attack method degrades the EAO scores from 0.287 to 0.093 (i.e., a 67.6% decrease) for SiamRPN++; from 0.283 to 0.115 (i.e., a 59.3% decrease) for SiamCAR; from 0.153 to 0.084 (i.e., a 45.1% decrease) for RT-MDNet; from 0.328 to 0.101 (i.e., a 69.2% decrease) for DiMP; and from 0.277 to 0.096 (i.e., a 65.3% decrease) for TransT. Compared to the attack by adding the same level of random noise, our attack method significantly degrades the tracking performance.

When applying our defense method, the performance of trackers is improved to 66.9% for SiamRPN++; 55.1% for SiamCAR; 72.5% for RT-MDNet; 53.0% for DiMP; and 50.5% for TransT of their original EAO scores. For VOT2018, the EAO scores are 0.082 for SiamRPN++; 0.104 for SiamCAR; 0.076 for RT-MDNet; 0.089 for DiMP; and 0.090 for TransT, when under attacks. By integrating our defense method, the EAO scores of SiamRPN++, SiamCAR, RT-MDNet, DiMP, and TransT are improved to 0.209, 0.136, 0.110, 0.203, and 0.121, respectively. For VOT2016, our adversarial attack algorithm reduces the EAO scores by over 60% for all five trackers. With the deployment of the defense method on the adversarial examples, the accuracy of EAO is improved to 39.7% for SiamRPN++; 53.8% for SiamRPN++; 60.8% for RT-MDNet; 56.6% for DiMP; and 50.7% for TransT of their original performance. The performance degradation and improvement indicate the effectiveness of our adversarial attack and defense methods on various trackers. Due to the reinitialization scheme in VOT, we observe

Table 1 Tracking performance of adversarial attack and defense methods on the VOT datasets (Kristan et al., 2019, 2018, 2016)

Dataset	VOT2019			VOT2018			VOT2016		
	Accuracy ↑	Robustness ↓	EAO ↑	Accuracy ↑	Robustness ↓	EAO ↑	Accuracy ↑	Robustness ↓	EAO ↑
SiamRPN++(Li et al., 2019)	0.594	0.467	0.287	**0.601**	0.234	**0.415**	**0.642**	0.196	0.464
SiamRPN++(Random)	0.574	0.625	0.236	0.581	0.412	0.284	0.637	0.298	0.360
SiamRPN++(Att)	0.502	2.017	0.093	0.500	1.681	0.082	0.485	1.664	0.084
SiamRPN++(Att+Def)	0.545	0.813	0.192	0.585	0.632	0.209	0.597	0.653	0.184
SiamRPN++(Def)	**0.596**	**0.426**	**0.294**	0.599	**0.225**	0.413	0.637	**0.158**	**0.472**
SiamCAR(Guo et al., 2020a)	0.593	0.461	0.283	0.589	0.281	0.354	0.633	0.219	0.420
SiamCAR(Random)	0.581	0.527	0.262	0.572	0.365	0.304	0.639	0.252	0.374
SiamCAR(Att)	0.570	2.047	0.115	0.574	1.723	0.104	0.602	1.221	0.136
SiamCAR(Att+Def)	**0.611**	1.344	0.156	**0.612**	1.199	0.136	**0.640**	0.662	0.226
SiamCAR(Def)	0.594	**0.431**	**0.294**	0.589	**0.258**	**0.375**	0.630	**0.186**	**0.431**
RT-MDNet(Jung et al., 2018)	0.527	0.873	0.153	**0.533**	0.567	0.176	**0.567**	0.196	0.370
RT-MDNet(Random)	0.523	1.144	0.139	0.503	0.871	0.137	0.550	0.452	0.235
RT-MDNet(Att)	0.462	1.986	0.084	0.475	1.611	0.076	0.469	0.928	0.128
RT-MDNet(Att+Def)	0.524	1.349	0.111	0.515	1.021	0.110	0.531	0.494	0.225
RT-MDNet(Def)	**0.551**	**0.863**	**0.168**	0.529	**0.538**	**0.179**	0.540	**0.168**	**0.374**
DiMP(Bhat et al., 2019)	0.561	0.302	0.328	0.575	0.174	0.412	**0.615**	0.145	0.461
DiMP(Random)	**0.572**	0.352	0.296	**0.584**	0.192	0.373	0.588	0.168	0.406
DiMP(Att)	0.506	1.766	0.101	0.519	1.527	0.089	0.548	1.678	0.077
DiMP(Att+Def)	0.569	0.833	0.174	0.584	0.553	0.203	0.603	0.405	0.261
DiMP(Def)	0.554	**0.277**	**0.336**	0.566	**0.155**	**0.430**	0.590	**0.140**	**0.470**
TransT(Chen et al., 2021)	**0.600**	0.502	0.277	0.596	0.370	0.287	0.634	0.252	0.367
TransT(Random)	0.595	0.627	0.244	**0.601**	0.482	0.243	**0.656**	0.396	0.294
TransT(Att)	0.595	2.212	0.096	0.586	1.709	0.090	0.621	1.142	0.136
TransT(Att+Def)	0.575	1.475	0.140	0.575	1.302	0.121	0.621	0.774	0.186
TransT(Def)	0.595	**0.451**	**0.287**	0.590	**0.332**	**0.292**	0.633	**0.242**	**0.380**

The best tracking performance results are given in bold
'Att' and 'Def' denote the adversarial attack and defense, respectively, and 'Random' denotes random perturbations

that EAO and robustness scores decrease dramatically during attacks, but the accuracy scores do not vary much. After applying our defense method, the robustness and EAO scores are primarily improved. Furthermore, when applying the proposed defense to the original sequences, our defense method slightly improves performance when the baseline trackers are not under attack.

4.4 Transferability Among Different Backbones

To evaluate the transferability among different backbone modules, we choose SiamRPN++ with AlexNet, MobileNet, and ResNet for experiments. Specifically, we generate adversarial examples from one backbone of SiamRPN++ and transfer them into the other two backbones. Furthermore, we conduct the experiments by injecting the same level of random noise for comparison. Table 2 shows the attack transferability for SiamRPN++ (Li et al., 2019) with AlexNet, MobileNet and ResNet on six benchmark datasets.

VOT datasets. Trackers with all backbones perform favorably on the original sequences. In VOT2019, SiamRPN++ with ResNet has a more complicated architecture than the other two networks, causing a 72.2% drop in EAO (i.e., from 0.291 to 0.081) on MobileNet and a 52.3% drop in EAO (i.e., from 0.260 to 0.124) on AlexNet. The perturbations from MobileNet and AlexNet have similar transferability on ResNet. In VOT2018, the transferable perturbations from ResNet degrade the performance of other backbones more significantly, which achieves an 84.1% decrease in EAO (i.e., from 0.410 to 0.065) for MobileNet and a 67.0% decrease in EAO (i.e., from 0.352 to 0.116) for AlexNet. In VOT2016, the perturbations from ResNet obtained over 50% drops in EAO for both AlexNet and MobileNet. In addition, the transferable adversarial perturbations from all backbones yield stronger attacks than random noises on all three VOT datasets.

Other datasets. Table 2 demonstrates the transferring attack results on the OTB100 (Wu et al., 2015), UAV123

Table 2 Transferability of the proposed method across different backbones on multiple datasets (Kristan et al., 2019, 2018, 2016; Wu et al., 2015; Mueller et al., 2016; Fan et al., 2019)

Dataset	Tracker	ResNet	MobileNet	AlexNet
VOT2019	Original	0.287	0.291	0.260
	Random	0.236	0.147	0.178
	ResNet	–	**0.081**	**0.124**
	MobileNet	0.165	–	0.132
	AlexNet	**0.163**	0.099	–
VOT2018	Original	0.415	0.410	0.352
	Random	0.284	0.137	0.189
	ResNet	–	**0.065**	**0.116**
	MobileNet	**0.155**	-	0.117
	AlexNet	0.158	0.086	–
VOT2016	Original	0.464	0.454	0.393
	Random	0.360	0.206	0.271
	ResNet	–	**0.090**	**0.192**
	MobileNet	**0.218**	-	0.202
	AlexNet	0.242	0.127	–
OTB100	Original	0.695	0.658	0.666
	Random	0.667	0.531	0.611
	ResNet	–	**0.251**	**0.546**
	MobileNet	0.531	–	0.571
	AlexNet	**0.527**	0.299	-
UAV123	Original	0.606	0.602	0.579
	Random	0.596	0.530	0.571
	ResNet	–	**0.288**	**0.476**
	MobileNet	0.532	–	0.505
	AlexNet	**0.504**	0.355	–
LaSOT	Original	0.494	0.450	0.434
	Random	0.462	0.319	0.392
	ResNet	–	**0.288**	**0.467**
	MobileNet	0.532	–	0.505
	AlexNet	**0.504**	0.355	–

The best attack results are given in bold

(Mueller et al., 2016) and LaSOT (Fan et al., 2019) datasets. In OTB100, the perturbations from MobileNet and AlexNet degrade from 0.695 to 0.531 and 0.527 on ResNet, respectively. The AUC scores of SiamRPN++ with MobileNet are reduced to 0.299 with AlexNet and 0.251 with ResNet. The transferring attack with MobileNet and ResNet decreases from 0.666 to 0.571 and 0.546 on AlexNet. In UAV123, the adversarial perturbations generated by ResNet also perform a more aggressive attack than the other two backbones, which yields a 52.2% drop (i.e., from 0.602 to 0.288) on MobileNet and a 17.8% drop (i.e., from 0.579 to 0.476) on AlexNet for AUC scores. In LaSOT, the AUC scores of all backbones drop by over 30% after applying our transferring attack. Generally, a backbone with complex network structures yields stronger attack transferability than simple structures. It can be explained by the fact that complex network structures (e.g., ResNet) learn more powerful representations. Our attack method is highly transferable to other backbones as a black-box attack and attacks the trackers more aggressively than the same level of random noise for all backbones.

4.5 Transferability Among Different Trackers

We conduct experiments to evaluate the transferability of our attack method among different targeted trackers. We use SiamRPN++ (Li et al., 2019) as our victim model and transfer the generated adversarial examples to other different types of trackers, SiamCAR (Guo et al., 2020a), RT-MDNet (Jung et al., 2018), DiMP (Bhat et al., 2019), and TransT (Chen et al., 2021). Similarly, we inject the same random noise level into the targeted trackers for comparisons. Table 3 reports the attack transferability across five diverse trackers on the six datasets (Kristan et al., 2016, 2018, 2019; Wu et al., 2015; Mueller et al., 2016; Fan et al., 2019).

VOT datasets. After transferring attack, the EAO scores in VOT2019 are reduced by 50.2% (i.e., from 0.283 to 0.141) for SiamCAR; 21.6% (i.e., from 0.153 to 0.120) for RT-MDNet; 48.6% (i.e., from 0.329 to 0.169) for DiMP; and 54.2% (i.e., from 0.277 to 0.127) for TransT. In VOT2018, the transferring attack from SiamRPN++ decreases from 0.354 to 0.128 (i.e., a 63.8% decrease) for SiamCAR; from 0.176 to 0.119 (i.e., a 32.4% decrease) for RT-MDNet; from 0.412 to 0.167 (i.e., a 59.5% decrease) for DiMP; and from 0.287 to 0.121 (i.e., a 57.8% decrease) for TransT, respectively. In VOT2016, the transferring attack from SiamRPN++ all yields a significant decrease of over 40% in EAO scores for SiamCAR, RT-MDNet, DiMP, and TransT. Despite the various architectures and training approaches of the targeted trackers, the adversarial examples generated by our proposed attack method can still degrade the tracking performance of these trackers noticeably. This suggests that our attack method exhibits substantial attack transferability across Siamese-based trackers, online-update trackers, discriminative trackers, and transformer trackers. Additionally, the attack performance is considerably better than the same level of random noise.

Other datasets. Different from the reinitialization scheme of VOT datasets, we utilize the AUC score as the evaluation metric to evaluate the attack transferability on the OTB100 (Wu et al., 2015), UAV123 (Mueller et al., 2016) and LaSOT (Fan et al., 2019) datasets. The AUC scores on OTB100 are reduced significantly from 0.697 to 0.352 for SiamCAR; from 0.643 to 0.516 for RT-MDNet; from 0.686 to 0.532 for DiMP, and from 0.690 to 0.577 for TransT. Our transferring attack on UAV123 degrades from 0.621 to 0.227 for SiamCAR; from 0.512 to 0.468 for RT-MDNet, from 0.646 to 0.528 for DiMP, and 0.679 from 0.551 for TransT. Also, the AUC scores on LaSOT drop from 0.516 to

Table 3 Transferability of the proposed method across different targeted trackers on multiple datasets (Kristan et al., 2019, 2018, 2016; Wu et al., 2015; Mueller et al., 2016; Fan et al., 2019)

Dataset	Method	SiamCAR	RT-MDNet	DiMP	TransT
VOT2019	Original	0.283	0.153	0.328	0.277
	Random	0.262	0.139	0.296	0.244
	SiamRPN++	**0.141**	**0.120**	**0.169**	**0.127**
VOT2018	Original	0.354	0.176	0.412	0.287
	Random	0.304	0.137	0.373	0.243
	SiamRPN++	**0.128**	**0.119**	**0.167**	**0.121**
VOT2016	Original	0.420	0.370	0.461	0.367
	Random	0.374	0.235	0.406	0.294
	SiamRPN++	**0.161**	**0.218**	**0.208**	**0.191**
OTB100	Original	0.697	0.643	0.686	0.690
	Random	0.679	0.559	0.674	0.654
	SiamRPN++	**0.325**	**0.516**	**0.532**	**0.577**
UAV123	Original	0.621	0.512	0.646	0.679
	Random	0.605	0.491	0.636	0.633
	SiamRPN++	**0.227**	**0.468**	**0.528**	**0.551**
LaSOT	Original	0.516	0.325	0.560	0.654
	Random	0.496	0.312	0.551	0.560
	SiamRPN++	**0.288**	**0.268**	**0.379**	**0.435**

The best attack results are given in bold

0.288 for SiamCAR; from 0.325 to 0.268 for RT-MDNet, from 0.560 to 0.379 for DiMP, and from 0.654 to 0.435 for TransT, respectively. The transferable perturbations from SiamRPN++ to SiamCAR exhibit better attack transferability compared to the other two trackers, as they share a similar architecture (i.e., Siamese-based tracker). Conversely, the perturbations from SiamRPN++ to RT-MDNet result in weak attacks due to the noticeable architecture gap between SiamRPN++ and RT-MDNet. While SiamRPN++ adopts a tracking-by-detection framework without online updates, RT-MDNet considers tracking a binary classification task and combines the online update module during inference. However, the generated perturbations from our method degrade the tracking performance considerably compared to the random noise. These results demonstrate the effectiveness of our transferring attack method as black-box attacks.

4.6 Universal Adversarial Perturbation

Although the proposed attack method significantly degrades tracker performance, the adversarial perturbation is computed based on the content of each frame, resulting in a high computational cost. A real-time attack speed can yield a more aggressive threat to real applications. Inspired by UAP (Moosavi-Dezfooli et al., 2017), we propose universal adversarial perturbations for tracking, which are dependent on the content and can be trained offline with training datasets. Once the universal adversarial perturbation is trained, it can be directly applied to all frames during inference. Therefore, the attack speed of universal attacks almost maintains the original tracking speed of trackers.

In the experiments, we use SiamRPN++ (Li et al., 2019) as a representative tracker to craft universal adversarial perturbations. We select 100 videos (∼32.4K frames) randomly from ImageNet VID (Russakovsky et al., 2015) and GOT-10k (Huang et al., 2019) as training data. Note that the variety of sampled training data has little impact on the attack performance of universal adversarial perturbations. We compare the attack performance of image-dependent perturbations (I-DP) and universal adversarial perturbations (UAP) on multiple datasets, as illustrated in Table 4. We utilize the normalized precision rate in the LaSOT dataset. Compared with the previous method, which generates individual perturbations for each frame, the attack performance of universal adversarial perturbations is slightly worse. However, they still significantly degrade the original tracking performance by a large margin. Specificially, the EAO scores on VOT2019 (Kristan et al., 2019), VOT2018 (Kristan et al., 2018), and VOT2016 (Kristan et al., 2016) decrease to 0.101, 0.092 and 0.150 respectively. After our UAP attack, the ACU scores on OTB100, UAV123, and LaSOT were reduced to 0.530, 0.543, and 0.286. Meanwhile, the universal attack speed is much faster than the image-dependent attack, as only additional operations are introduced during inference.

To analyze the attack effects between I-DP and UPA, we define "completely lose tracking" as the occurrence of five consecutive frames in which the predicted boxes have no overlap with the ground truth bounding boxes. The rea-

Table 4 Tracking performance with universal adversarial perturbations on multiple datasets (Kristan et al., 2019, 2018, 2016; Wu et al., 2015; Mueller et al., 2016; Fan et al., 2019)

Dataset	Method	Accuracy ↑	Robustness ↑	EAO ↑
VOT2019	Original	0.594	0.467	0.287
	I-DP	**0.502**	**2.017**	**0.093**
	UAP	0.545	1.926	0.101
VOT2018	Original	0.601	0.234	0.415
	I-DP	**0.500**	**1.681**	**0.082**
	UAP	0.551	1.587	0.092
VOT2016	Original	0.642	0.196	0.464
	I-DP	**0.485**	**1.664**	**0.084**
	UAP	0.573	0.965	0.150
Dataset	Method	Succ. ↑	Prec. ↑	Speed ↑
OTB100	Original	0.695	0.905	75.3
	I-DP	**0.038**	**0.033**	1.8
	UAP	0.401	0.530	**74.1**
UAV123	Original	0.606	0.798	75.8
	I-DP	**0.043**	**0.069**	1.7
	UAP	0.409	0.543	**74.7**
LaSOT	Original	0.496	0.575*	75.3
	I-DP	**0.038**	0.017*	1.7
	UAP	0.247	0.286*	74.4

The best attack results are given in bold

Table 5 The averaged frame for completely lose tracking on the OTB100 (Wu et al., 2015), UAV123 (Mueller et al., 2016) and LaSOT (Fan et al., 2019) datasets

Dataset	OTB100	UAV123	LaSOT
I-DP	34.12	87.22	60.70
UAP	316.31	507.01	606.51

son for choosing five consecutive frames is that when the overlap ratio reaches zero initially, the tracker will recover its performance if there are no adversarial perturbations in the following frames. We count the completely lost tracking frame for independent and universal attacks and report the average number of frames in various datasets in Table 5. We observe that the image-dependent attacks have more attacking strength with lower attack speeds than the universal adversarial attacks.

4.7 Ablation Studies

We analyze the effectiveness of each component of the proposed method in this section. Specifically, we use SiamRPN++ (Li et al., 2019) as a baseline and conduct ablation studies on the OTB100 (Wu et al., 2015) and UAV123 (Mueller et al., 2016) dataset.

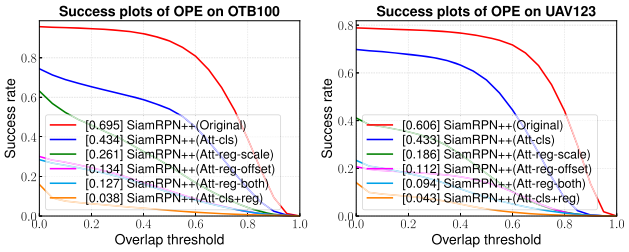


Fig. 7 Variants of adversarial attack on the OTB100 (Wu et al., 2015) and UAV123 (Mueller et al., 2016) dataset

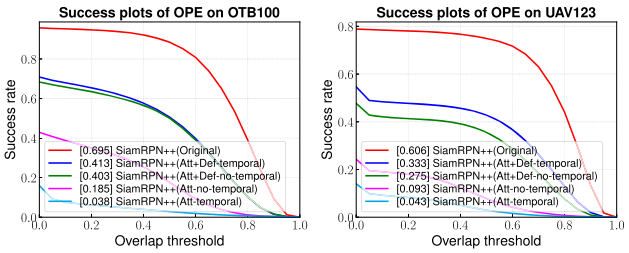


Fig. 8 Temporal consistency validation of adversarial attack and defense on the OTB100 (Wu et al., 2015) and UAV123 (Mueller et al., 2016) dataset

Network branches. SiamRPN++ contains both the classification and regression branches. We first evaluate the tracking performance on the original sequences as a baseline. Then, we separately apply our attack method to the classification and regression branches. When we attack the regression branch, we analyze the offset and scale variation effects. Finally, we combine both classification and regression attacks. Note that we denote cls as the attack on the classification branch and reg as the attack on the regression branch where offset and scale attacks exist. Figure 7 shows the evaluation results where the attack on the regression branch degrades the performance more significantly than the attack on the classification branch. Combining attacks on both branches yields the most significant degradation in tracking performance.

Temporal consistency. We compare our attack and defense with and without temporal consistency in Fig. 8. Our temporal attack decreases the tracking accuracy more heavily compared to the adversarial attack only on static images. Meanwhile, the temporal initialization benefits the strength of our attack more than our defense.

Sign function. We utilize the sign function for attack and remove it for defense. In Fig. 9, we compare our attack and defense with and without the sign function. The results indicate that using the sign function or not for attacks has no obvious effects on attack performance. However, removing the sign function during defense yields better defense results than retaining it, which illustrates the effectiveness of our defense by removing the sign function.

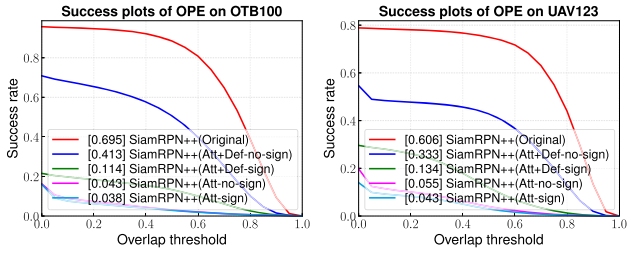


Fig. 9 Sign function validation of adversarial attack and defense on the OTB100 (Wu et al., 2015) and UAV123 (Mueller et al., 2016) dataset

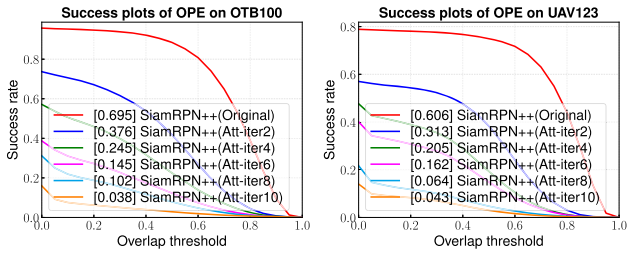


Fig. 10 Performance of different iterations M on the OTB100 (Wu et al., 2015) and UAV123 (Mueller et al., 2016) dataset

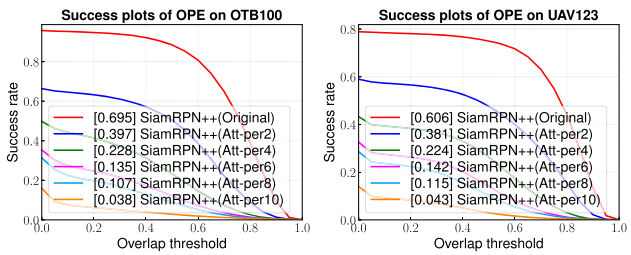


Fig. 11 Performance of different maximal value ϵ of perturbations on the OTB100 (Wu et al., 2015) and UAV123 (Mueller et al., 2016) dataset

Iterations. We report the attack performance by increasing the iteration number M. Figure 10 shows the variations of success rates with different iterations M. Like other attacks, more iterations yield robust attack performance but cost computation and time.

Maximal value of perturbations. To illustrate the effect of different magnitudes of perturbations, we limit the maximal variation on each pixel for our attack. Figure 11 reports the success rates with different maximum values ϵ. Intuitively, more significant perturbations cause larger performance drops. We choose the maximal variation of perturbations ϵ as 10 for adversarial attacks, which are still not noticeable to human observers.

4.8 Comparisons with Other Methods

We evaluate the proposed attack and defense against other schemes under the same experimental settings for fair comparisons. We apply our attack method on SiamRPN++ (Li et al., 2019) with ResNet as an illustration.

Existing adversarial attacks. The comparison with existing adversarial attacks is reported in Table 6. In the case of the white-box attack, we initially compare our attack with the attack methods against object detection, such as DAG (Xie et al., 2017b) and Daedalus (Wang et al., 2021). These methods lack the temporal attack, which results in limited attack performance. We evaluate our attack with the attack methods against tracking, which can be categorized as multi-object tracking and single-object tracking. Adversarial attacks for multiple object tracking (MOT) methods have been analyzed in recent years (Lin et al., 2021), (Zhou et al., 2023; Jia et al., 2019). TraSw (Lin et al., 2021) and F&F attack (Zhou et al., 2023) focus on misleading the association process or crafting unreliable detection boxes, which are not applicable to our single object tracing task with only a single box per frame. Hijacking (Jia et al., 2019) aims to attack detection boxes and data association simultaneously but yields limited performance when transferred to single object tracking. For single object tracking, we compare our attack with Ad^2 attack (Fu et al., 2022), CSA (Yan et al., 2020), SPARK (Guo et al., 2020b), One-shot (Chen et al., 2020a) and ABA (Guo et al., 2021) under the same setting. All these methods significantly degrade the trackers, while our approach degrades the tracker at the first few frames, leading to a dramatic drop (i.e., 0.658 success rates and 0.878 precision rates).

On the other hand, we also compare the attack performance under the black-box setting. Concretely, we use SiamRPN++ with ResNet as the targeted tracker. We implement the same experiments for SPARK and our attack by transferring the adversarial perturbations generated by the white-box tracker to the targeted tracker. The attack results show that our attack has better attack transferability than SPARK, as our method attacks both classification and regress modules and involves the temporal coherent information between frames during attacks. Furthermore, we compare the proposed method with a decision-based black-box attack by IoU attack (Jia et al., 2021). Our attack is slightly worse than IoU attack, which constantly queries the targeted tracker to adjust the direction of the attack and consumes heavy computation. However, our attack without access to the targeted tracker directly transfers the generated adversarial perturbations with a more rapid attack speed.

Existing adversarial defenses. We compare our defense with common defense algorithms, including adversarial training, spatial smoothing, and color jittering. For spatial smoothing, we utilize Gaussian blur as an illustration. We choose random color jittering, including brightness, contrast, saturation, and hue. The defense results are reported in Table 7. It indicates that all defense methods partially restore the tracking performance. However, the spatial smoothing

Table 6 Comparison with existing adversarial attacks on the OTB100 (Wu et al., 2015) dataset

Type	Attack method	Succ. Drop	Prec. Drop
Detection (White-box)	DAG (Xie et al., 2017b)	0.147	0.182
	Daedalus (Wang et al., 2021)	0.201	0.226
Tracking (White-box)	Hijacking (Jia et al., 2019)	0.238	0.216
	Ad ^{2} attack (Fu et al., 2022)	0.289	0.376
	CSA (Yan et al., 2020)	0.372	0.443
	SPARK (Guo et al., 2020b)	0.223	0.333
	One-shot (Chen et al., 2020a)	0.444	0.577
	ABA (Guo et al., 2021)	0.312	0.417
	Ours	**0.658**	**0.878**
Tracking (Black-box)	SPARK (Guo et al., 2020b)	0.066	0.027
	IoU attack (Jia et al., 2021)	**0.196**	**0.261**
	Ours	0.168	0.208

The best attack results are given in bold

Table 7 Comparison with common adversarial defenses on the OTB100 (Wu et al., 2015) dataset

Defense Method	Success	Precision
SiamRPN++ (Original)	0.695	0.905
SiamRPN++ (Attack)	0.038	0.033
Adversarial Training	0.268	0.318
Spatial smoothing (Gaussian)	0.340	0.423
Color jittering	0.183	0.252
Ours	0.413	0.535

Table 8 Comparsion with the proposed defense against various attacks on the OTB100 (Wu et al., 2015) dataset

Attack Method	Success	Precision
CSA (Yan et al., 2020) (Att)	0.346	0.489
CSA (Yan et al., 2020) (Att+Def)	0.413	0.588
SPARK (Guo et al., 2020b) (Att)	0.473	0.575
SPARK (Guo et al., 2020b) (Att+Def)	0.620	0.752
IoU Attack (Jia et al., 2021) (Att)	0.499	0.644
IoU Attack (Jia et al., 2021) (Att+Def)	0.538	0.724

and color jittering could affect the image quality, while adversarial training is time-consuming as it requires re-training of the trackers. In contrast, our proposed defense outperforms the other defense methods and maintains the original image quality considerably.

Our defense against other attacks. To demonstrate the generality of our defense against various attacks, we choose gradient-based white-box attacks (Yan et al., 2020; Guo et al., 2020b) and decision-based black-box attacks (Jia et al., 2021) for evaluations. The experiment results against other attacks with the proposed defense are presented in Table 8. Since the proposed attack is designed to defend our gradient-based attack, it can effectively protect against other gradient-based attacks (i.e., CSA (Chen et al., 2020a) and SPARK (Guo et al., 2020b)), but its performance is limited against decision-based attacks (i.e., IoU Attack (Jia et al., 2021)). We will further investigate the generalized defense in future works.

4.9 Limitations

We summarize the limitations of our method and potential directions for improvement in our future work. First, although the proposed attack method considerably degrades the tracking performance compared to white-box attacks, the transferability of generated adversarial perturbation compared to black-box attacks has much room to improve. This is especially true when transferring between two trackers with significant differences in architecture (e.g., SiamRPN++ (Li et al., 2019) and RT-MDNet (Jung et al., 2018)). Our future work will explore a unified representation of features for various trackers, further strengthening attack transferability across multiple trackers. Second, since our defense method relies on trackers' concrete architecture, we need to manually set these empirical weights when defending the attack on different trackers. This suggests learning a dynamic parameter adjustment according to diverse trackers and adversarial attacks in our future work.

5 Conclusion

In this paper, we explore the adversarial attack and defense for visual object tracking. We propose an adversarial attack to generate lightweight perturbations on the original video sequences. When crafting adversarial examples, we integrate temporal perturbations into frames by perplexing trackers

with indistinguishable correct and incorrect inferences as white-box attacks. For black-box attacks, we transfer the adversarial perturbations generated by the victim tracker to other unknown backbones and trackers, dramatically dropping their performance. Furthermore, we train universal adversarial perturbations to add them into all frames, which significantly degrades the tracking performance and improves the attack speed. When defending adversarial examples, we suppress the maximum of adversarial perturbations to restore tracking accuracy. Extensive experiments on six benchmark datasets demonstrate that the proposed methods perform favorably in attack and defense. In addition, our defense method can reduce interference from perturbations in real-world scenarios to robustify deep trackers.

Acknowledgements This work was supported in part by NSFC (62322113, 62376156), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and the Fundamental Research Funds for the Central Universities.

References

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., & Torr, P. H. (2016). Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision (ECCV)* .

Bhat, G., Danelljan, M., Van Gool, L., & Timofte, R. (2019). Learning discriminative model prediction for tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Borsuk, V., Vei, R., Kupyn, O., Martyniuk, T., Krashenyi, I., & Matas, J. (2022). Fear: Fast, efficient, accurate and robust visual tracker. In *European conference on computer vision (ECCV)*.

Brendel, W., Rauber, J., & Bethge, M. (2018). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International conference on learning representations (ICLR)*.

Cao, Z., Fu, C., Ye, J., Li, B., & Li, Y. (2021). Hift: Hierarchical feature transformer for aerial tracking. In *IEEE/CVF international conference on computer vision (ICCV)*.

Chen, B., Li, P., Bai, L., Qiao, L., Shen, Q., Li, B., Gan, W., Wu, W., & Ouyang, W. (2022). Backbone is all your need: A simplified architecture for visual object tracking. In *European conference on computer vision (ECCV)*.

Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., & Lu, H. (2021). Transformer tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Chen, X., Yan, X., Zheng, F., Jiang, Y., Xia, S. T., Zhao, Y., & Ji, R. (2020). One-shot adversarial attacks on visual tracking with dual attention. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Chen, Z., Zhong, B., Li, G., Zhang, S., & Ji, R. (2020). Siamese box adaptive network for visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Cheng, M., Le, T., Chen, P. Y., Zhang, H., Yi, J., & Hsieh, C. J. (2018). Query-efficient hard-label black-box attack: An optimization-based approach. In *International conference on learning representations (ICLR)*.

Cui, Y., Jiang, C., Wang, L., & Wu, G. (2022) Mixformer: End-to-end tracking with iterative mixed attention. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Dai, K., Zhang, Y., Wang, D., Li, J., Lu, H., & Yang, X. (2020). High-performance long-term tracking with meta-updater. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Danelljan, M., Bhat, G., Khan, F. S., & Felsberg, M. (2019). Atom: Accurate tracking by overlap maximization. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Danelljan, M., Bhat, G., Khan, F. S., & Felsberg, M. (2017). ECO: Efficient convolution operators for tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Danelljan, M., Robinson, A., Khan, F. S., & Felsberg, M. (2016). Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European conference on computer vision (ECCV)*.

Ding, L., Wang, Y., Yuan, K., Jiang, M., Wang, P., Huang, H., & Wang, Z. J. (2021). Towards universal physical attacks on single object tracking. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*.

Dong, Y., Pang, T., Su, H., & Zhu, J. (2019). Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., & Zhu, J. (2019). Efficient decision-based black-box adversarial attacks on face recognition. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S., & Uszkoreit, J. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations (ICLR)*.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., & Song, D. (2018). Robust physical-world attacks on deep learning models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., & Ling, H. (2019). Lasot: A high-quality benchmark for large-scale single object tracking. In *IEEE/CVF international conference on computer vision (ICCV)*.

Fu, C., Li, S., Yuan, X., Ye, J., Cao, Z., & Ding, F. (2022). Ad2Attack: Adaptive adversarial attack on real-time UAV tracking. In *2022 international conference on robotics and automation (ICRA)*.

Gao, S., Zhou, C., Ma, C., Wang, X., & Yuan, J. (2022). Aiatrack: Attention in attention for transformer visual tracking. In *European conference on computer vision (ECCV)*.

Gao, S., Zhou, C., & Zhang, J. (2023) Generalized relation modeling for transformer tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International conference on learning representations (ICLR)*.

Guo, C., Rana, M., Cisse, M., & Van Der Maaten, L. (2018). Countering adversarial images using input transformations. In *International conference on learning representations (ICLR)*.

Guo, D., Wang, J., Cui, Y., Wang, Z., & Chen, S. (2020). Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Guo, Q., Cheng, Z., Juefei-Xu, F., Ma, L., Xie, X., Liu, Y., & Zhao, J. (2021). Learning to adversarially blur visual object tracking. In *IEEE/CVF international conference on computer vision (ICCV)*.

Guo, Q., Xie, X., Juefei-Xu, F., Ma, L., Li, Z., Xue, W., Feng, W., & Liu, Y. (2020). Spark: Spatial-aware online incremental attack against visual tracking. In *European conference on computer vision (ECCV)*.

Gupta, K., Pesquet-Popescu, B., Kaakai, F., Pesquet, J. C., & Malliaros, F.D. (2021). An adversarial attacker for neural networks in regres-

sion problems. In *IJCAI workshop on artificial intelligence safety (AI Safety)*.

Han, B., Sim, J., & Adam, H. (2017). Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Held, D., Thrun, S., & Savarese, S. (2016). Learning to track at 100 fps with deep regression networks. In *European conference on computer vision (ECCV)*.

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

Ilyas, A., Engstrom, L., Athalye, A., & Lin, J. (2018). Black-box adversarial attacks with limited queries and information. In *International conference on machine learning (ICML)*.

Jia, S., Ma, C., Song, Y., & Yang, X. (2020). Robust tracking against adversarial attacks. In *European conference on computer vision (ECCV)*.

Jia, S., Song, Y., Ma, C., & Yang, X. (2021). Iou attack: Towards temporally coherent black-box adversarial attack for visual object tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Jia, Y., Lu, Y., Shen, J., Chen, Q. A., Zhong, Z., & Wei, T. (2019). Fooling detection alone is not enough: First adversarial attack against multiple object tracking. In *International conference on learning representations (ICLR)*.

Jung, I., Son, J., Baek, M., & Han, B. (2018). Real-time MDnet. In *European conference on computer vision (ECCV)*.

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., Vojir, T., Bhat, G., Lukezic, A., Eldesokey, A., Fernandez, G.(2018). The sixth visual object tracking vot2018 challenge results. In *European conference on computer vision (ECCV) workshop*.

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., Vojir, T., Hager, G., Lukezic, A., Eldesokey, A. (2016). The visual object tracking vot2016 challenge results. In *European conference on computer vision (ECCV) workshop*.

Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J. K., Cehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., & Eldesokey, A. (2019). The seventh visual object tracking vot2019 challenge results. In *European conference on computer vision (ECCV) workshop*.

Kurakin, A., Goodfellow, I., & Bengio, S. (2017). Adversarial examples in the physical world. In *International conference on learning representations (ICLR)*.

Lai, P., Cheng, G., Zhang, M., Ning, J., Zheng, X., & Han, J. (2023). Ncsiam: Reliable matching via neighborhood consensus for siamese-based object tracking. *IEEE Transactions on Image Processing, 32*, 6168.

Lai, P., Zhang, M., Cheng, G., Li, S., Huang, X., & Han, J. (2023). Target-aware transformer for satellite video object tracking. *IEEE Transactions on Geoscience and Remote Sensing, 62*, 1–10.

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., & Yan, J. (2019). Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Li, B., Yan, J., Wu, W., Zhu, Z., & Hu, X. (2018). High performance visual tracking with siamese region proposal network. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Li, F., Tian, C., Zuo, W., Zhang, L., & Yang, M. H. (2018). Learning spatial-temporal regularized correlation filters for visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Liang, S., Wei, X., Yao, S., & Cao, X. (2020). Efficient adversarial attacks for visual object tracking. In *European conference on computer vision (ECCV)*.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., & Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Lin, L., Fan, H., Zhang, Z., Xu, Y., & Ling, H. (2022). Swintrack: A simple and strong baseline for transformer tracking. *Advances in Neural Information Processing Systems, 35*, 16743.

Liu, Y., Chen, X., Liu, C., & Song, D. (2017). Delving into transferable adversarial examples and black-box attacks. In *International conference on learning representations (ICLR)*.

Lu, J., Sibai, H., & Fabry, E. (2017). Adversarial examples that fool detectors. arXiv preprint arXiv:1712.02494

Lin, D., Chen, Q., Zhou, C., & He, K. (2021). TraSw: Tracklet-switch adversarial attacks against multi-object tracking. https://doi.org/10.31219/osf.io/tdc9b

Lu, X., Ma, C., Ni, B., Yang, X., Reid, I., & Yang, M. H. (2018). Deep regression tracking with shrinkage loss. In *European conference on computer vision (ECCV)*.

Ma, C., Huang, J. B., Yang, X., & Yang, M. H. (2015). Hierarchical convolutional features for visual tracking. In *IEEE/CVF international conference on computer vision (ICCV)*.

Ma, F., Shou, M.Z., Zhu, L., Fan, H., Xu, Y., Yang, Y., & Yan, Z. (2022). Unified transformer tracker for object tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International conference on learning representations (ICLR)*.

Mayer, C., Danelljan, M., Bhat, G., Paul, M., Paudel, D. P., Yu, F., & Van Gool, L. (2022). Transforming model prediction for tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Moosavi-Dezfooli, S. M., Fawzi, A., Fawzi, O., & Frossard, P. (2017). Universal adversarial perturbations. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). Deepfool: A simple and accurate method to fool deep neural networks. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Mueller, M., Smith, N., & Ghanem, B. (2016). A benchmark and simulator for UAV tracking. In *European conference on computer vision (ECCV)*.

Nam, H., & Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Pu, S., Song, Y., Ma, C., Zhang, H., & Yang, M. H. (2018). Deep attentive tracking via reciprocative learning. In *Advances in neural information processing systems (NeurIPS)*.

Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., & Yang, M. H. (2016). Hedged deep tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NeurIPS)*.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*, 1137.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., & Berg, A. C.

(2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*, 211.

Shen, Q., Qiao, L., Guo, J., Li, P., Li, X., Li, B., Feng, W., Gan, W., Wu, W., & Ouyang, W. (2022). Unsupervised learning of accurate siamese tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R.W., & Yang, M. H. (2017). CREST: Convolutional residual learning for visual tracking. In *IEEE/CVF international conference on computer vision (ICCV)*.

Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., Shen, C., Lau, R.W., & Yang, M.H. (2018). VITAL: Visual tracking via adversarial learning. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Song, Z., Yu, J., Chen, Y. P. P., & Yang, W. (2022). Transformer tracking with cyclic shifting window attention. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Sun, B., Tsai, N.h., Liu, F., Yu, R., & Su, H. (2019). Adversarial defense by stratified convolutional sparse coding. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Sun, X., Cheng, G., Li, H., Pei, L., Han, J. (2022). Exploring effective data for surrogate training towards black-box attack. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Sun, X., Cheng, G., Li, H., Pei, L., & Han, J. (2023). On single-model transferable targeted attacks: A closer look at decision-level optimization. *IEEE Transactions on Image Processing, 32*, 2972.

Sun, Y., Sun, C., Wang, D., He, Y., & Lu, H. (2019). Roi pooled correlation filters for visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In *International conference on learning representations (ICLR)*.

Tramèr, F., Boneh, D., Kurakin, A., Goodfellow, I., Papernot, N., & McDaniel, P. (2018). Ensemble adversarial training: Attacks and defenses. In *International conference on learning representations (ICLR)*.

Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., & Torr, P. H. (2017). End-to-end representation learning for correlation filter based tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*.

Wang, D., Li, C., Wen, S., Han, Q. L., Nepal, S., Zhang, X., & Xiang, Y. (2021). Daedalus: Breaking nonmaximum suppression in object detection via adversarial examples. *IEEE Transactions on Cybernetics, 52*, 7427.

Wang, L., Ouyang, W., Wang, X., & Lu, H. (2015). Visual tracking with fully convolutional networks. In *IEEE/CVF international conference on computer vision (ICCV)*.

Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., & Li, H. (2019). Unsupervised deep tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Wang, N., Zhou, W., Song, Y., Ma, C., Liu, W., & Li, H. (2020). Unsupervised deep representation learning for real-time tracking. *International Journal of Computer Vision, 129*, 400.

Wang, N., Zhou, W., Wang, J., & Li, H. (2021). Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Wang, W., Yin, B., Yao, T., Zhang, L., Fu, Y., Ding, S., Li, J., Huang, F., Xue, X. (2021). Delving into data: Effectively substitute training for black-box attack. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Wei, X., Bai, Y., Zheng, Y., Shi, D., & Gong, Y. (2023). Autoregressive visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Wiyatno, R. R., & Xu, A. (2019). Physical adversarial textures that fool visual object tracking. In *IEEE/CVF international conference on computer vision (ICCV)*.

Wu, Y., Lim, J., & Yang, M. H. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xiao, C., Deng, R., Li, B., Yu, F., Liu, M., & Song, D. (2018). Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In *European conference on computer vision (ECCV)*.

Xie, C., Wang, J., Zhang, Z., Ren, Z., & Yuille, A. (2017). Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., & Yuille, A. (2017). Adversarial examples for semantic segmentation and object detection. In *IEEE/CVF international conference on computer vision (ICCV)*.

Xie, C., Wu, Y., van der Maaten, L., Yuille, A. L., & He, K. (2019). Feature denoising for improving adversarial robustness. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Xie, F., Wang, C., Wang, G., Cao, Y., Yang, W., & Zeng, W. (2022). Correlation-aware deep tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Xie, F., Wang, C., Wang, G., Yang, W., & Zeng, W. (2021). Learning tracking representations via dual-branch fully transformer networks. In *IEEE/CVF international conference on computer vision (ICCV)*.

Xing, D., Evangeliou, N., Tsoukalas, A., & Tzes, A. (2022). Siamese transformer pyramid networks for real-time UAV tracking. In *IEEE/CVF winter conference on applications of computer vision (WACV)*.

Yan, B., Jiang, Y., Sun, P., Wang, D., Yuan, Z., Luo, P., & Lu, H. (2022). Towards grand unification of object tracking. In *European conference on computer vision (ECCV)*.

Yan, B., Peng, H., Fu, J., Wang, D., & Lu, H. (2021). Learning spatio-temporal transformer for visual tracking. In *IEEE/CVF international conference on computer vision (ICCV)*.

Yan, B., Wang, D., Lu, H.,& Yang, X. (2020). Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Yu, B., Tang, M., Zheng, L., Zhu, G., Wang, J., Feng, H., Feng, X., & Lu, H. (2021). High-performance discriminative tracking with transformers. In *IEEE/CVF international conference on computer vision (ICCV)*.

Zhang, L., Gonzalez-Garcia, A., Weijer, J. V. D., Danelljan, M., & Khan, F. S. (2019). Learning the model update for siamese trackers. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Zhang, T., Xu, C., & Yang, M. H. (2017) Multi-task correlation particle filter for robust object tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Zhang, Z., & Peng, H. (2019). Deeper and wider siamese networks for real-time visual tracking. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Zhang, Z., Peng, H., Fu, J., Li, B., & Hu, W. (2020). Ocean: Object-aware anchor-free tracking. In *European conference on computer vision (ECCV)*.

Zhou, M., Wu, J., Liu, Y., Liu, S., & Zhu, C. (2020). Dast: Data-free substitute training for adversarial attacks. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Zhou, T., Ye, Q., Luo, W., Zhang, K., Shi, Z., & Chen, J. (2023). F&f attack: Adversarial attack against multiple object trackers by inducing false negatives and false positives. In *IEEE/CVF international conference on computer vision (ICCV)*.

Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., & Hu, W. (2018). Distractor-aware siamese networks for visual object tracking. In *European conference on computer vision (ECCV)*.