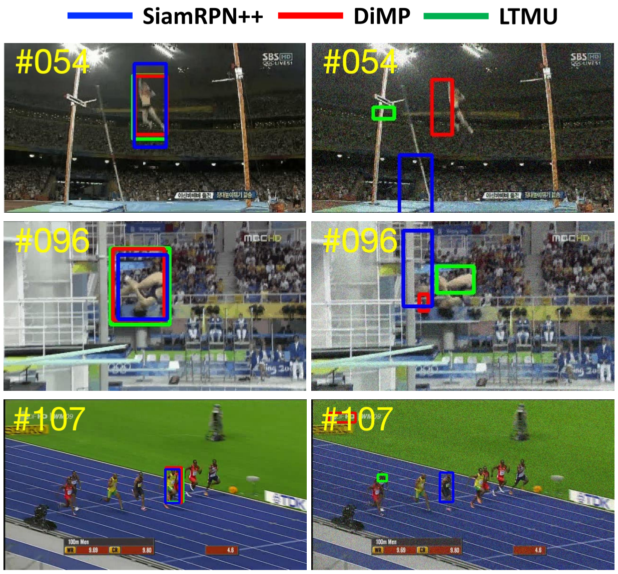
IoU Attack: Towards Temporally Coherent Black-Box Adversarial Attack for Visual Object Tracking

Shuai Jia[1] Yibing Song[2] Chao Ma[1]* Xiaokang Yang[1]
[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]Tencent AI Lab
{jiashuai,chaoma,xkyang}@sjtu.edu.cn, yibingsong.cv@gmail.com

Abstract

Adversarial attack arises due to the vulnerability of deep neural networks to perceive input samples injected with imperceptible perturbations. Recently, adversarial attack has been applied to visual object tracking to evaluate the robustness of deep trackers. Assuming that the model structures of deep trackers are known, a variety of white-box attack approaches to visual tracking have demonstrated promising results. However, the model knowledge about deep trackers is usually unavailable in real applications. In this paper, we propose a decision-based black-box attack method for visual object tracking. In contrast to existing black-box adversarial attack methods that deal with static images for image classification, we propose IoU attack that sequentially generates perturbations based on the predicted IoU scores from both current and historical frames. By decreasing the IoU scores, the proposed attack method degrades the accuracy of temporal coherent bounding boxes (i.e., object motions) accordingly. In addition, we transfer the learned perturbations to the next few frames to initialize temporal motion attack. We validate the proposed IoU attack on state-of-the-art deep trackers (i.e., detection based, correlation filter based, and long-term trackers). Extensive experiments on the benchmark datasets indicate the effectiveness of the proposed IoU attack method. The source code is available at https://github.com/VISION-SJTU/IoUattack.



(a) Original results (b) IoU attack

Figure 1. IoU attack for visual object tracking. State-of-the-art deep trackers (i.e., SiamRPN++ [22], DiMP [1], and LTMU [5]) effectively locate target objects in the original video sequences as shown in (a). Our IoU attack decreases their tracking accuracies by injecting imperceptible perturbations as shown in (b).

1. Introduction

Visual object tracking is one of the fundamental computer vision problems with a wide range of applications. The convolutional neural networks (CNNs) have significantly advanced visual tracking performance. Meanwhile, the enigma of interpreting CNNs has perplexed existing visual tracking algorithms as well. For example, injecting imperceptible perturbations into input images leads deep neural networks to predict incorrectly [37, 43, 48]. To investigate the robustness of visual tracking algorithms with deep models, recent approaches [3, 44, 16, 24] assume that the model structures of deep tracking algorithms are known and carry out white-box attack on them. Despite the demonstrated promising results, the concrete structures and parameters of deep trackers are barely known in real applications. In this paper, we investigate black-box adversarial attack for visual tracking, where the model knowledge of deep trackers is unknown.

Prevalent black-box attack algorithms inject imperceptible perturbations into input images to decrease network classification accuracies. Although these methods are effective to attack static images, they are not suitable to attack temporally moving objects in videos. This is because

*Corresponding author.

deep trackers maintain temporal motions of the target object within tracking models (i.e., the correlation filters [6, 35] or deep binary classifiers [30, 17, 23, 22]). When localizing the target object, these deep trackers produce temporally coherent bounding boxes (bbxs). Meanwhile, deep trackers constrain the search area to be close to the predicted bbx from the last frame. As existing black-box methods rarely degrade temporally coherent bbxs, perturbations produced based on CNN classification scores are not effective for visual tracking. An intriguing direction thus arises to investigate the black-box attack on both individual frames and temporal motions among sequential frames with a holistic decision-based approach.

In this paper, we propose IoU attack for visual tracking. IoU attack is a decision-based black-box attack method which focuses on both image content and target motions in video sequences. When processing each frame, we start image content attack with two bbxs. One is predicted by the deep tracker using the original frame, which is perturbation free. The other one is predicted by the same tracker using the same frame with noisy perturbations. These two bbxs are used to compute an IoU score as feedback to our IoU attack. For each frame, we use an iterative orthogonal composition method for image content attack. During each iteration of orthogonal attack, we first randomly generate several tangential perturbations whose noise levels are the same. Then, we compute their IoU scores and select the tangential perturbation with the lowest score. The selected perturbation is the most effective one to attack the current frame at the current iteration. We then increase the selected perturbation in its normal direction to add a small amount of noise, which is the normal perturbation. We compose both tangential and normal perturbations to generate the perturbations for the current iteration of orthogonal attack.

For target motion attack, we compute an IoU score between the bbxs from both the current and the previous frames. This IoU score is integrated into the tangential perturbation identification process. To this end, our orthogonal attack deviates a deep tracker from its original performance of both the current and historical frames. We transfer the learned perturbations to the next few frames as perturbation initialization to reinforce temporal motion attack. As a result, the deviation from the original tracking results ensures the success of black-box attack on deep trackers shown in Figure 1. We extensively validate the proposed IoU attack on state-of-the-art methods including detection based [22], correlation filter based [1], and long-term [5] trackers. Experiments on benchmark datasets demonstrate the effectiveness of the proposed black-box IoU attack.

2. Related Work

In this section, we briefly introduce recent state-of-the-art trackers and their basic principles. Besides, we also review recent adversarial attack methods, especially for the aspect of black-box attack.

2.1. Visual Object Tracking

Visual object tracking has received widespread attention in the last decade and brings about a series of new benchmark datasets [42, 18, 28, 29, 11]. Existing trackers can be generally categorized as offline trackers and online update trackers. Offline trackers do not update their model parameters during the inference, leading to a higher speed. These trackers consider tracking as a discriminative object detection problem. They generate candidate regions and classify the target or background to locate. Bounding box regression [34] is always used to locate precisely. Among them, siamese based methods [23, 22, 39, 47, 13, 4, 38, 40] are typical structures consisting of a template branch and a search branch. SiamRPN [23] draws a region proposal network to formulate a one-shot detection by comparing the similarity between two branches. SiamRPN++ [22] applies a deeper network ResNet instead of commonly-used AlexNet to improve the tracking accuracy and maintain the real-time speed.

Online update trackers constantly update their models during the inference to adapt to the current scenarios [31]. MDNet [30, 36, 32] regard tracking as a classification to distinguish the target and background. During the inference, they collect the samples from previous frames to enhance the target appearance. UpdateNet [46] formulates an update strategy into siamese based trackers to maintain the temporal motion between frames. Besides, correlation filter based methods also belong to online update trackers. They typically learn the discriminative correlation filter by deep or hand-craft features to estimate the target location. Recently, DiMP [1] learns a discriminative learning loss to exploit both target and background appearance information for target model prediction. PrDiMP [7] proposes a probabilistic regression formulation to address the modeling label noise.

Furthermore, existing long-term trackers [45, 5] integrate an online update module to improve the tracking performance. The re-detection module is mostly introduced to handle the disappearance and reappearance of the target, involving more challenges into adversarial attack. LTMU [5] is a long-term tracker with a meta-updater, which learns to guide the tracker's update to gain helpful appearance information for accuracy. In this work, we implement our adversarial attack on three representative trackers [22, 1, 5] to illustrate the generality of our black-box attack method.

2.2. Adversarial Attack

Convolution Neural Networks (CNNs) have been deployed in various tasks of computer vision today. However, recent studies [12, 37] notice that CNNs are sensitive to the imperceptible perturbations in adversarial examples. The intentional light-weight perturbations deteriorate the performance dramatically. Existing adversarial attack methods [12, 27, 15, 8, 33] mainly focus on static image tasks like classification, segmentation and detection. Except for attacking digital images, some studies implement physical attacks [10, 41] in concrete applications (e.g., autonomous driving). They generate a distractor in the real world to cause CNNs models to misclassify or fail to detect, leading to a security problem.

Overall, existing adversarial attack methods are mainly divided into two categories: white-box and black-box attack. In white-box attack, the adversary assumes to gain all knowledge of the attacked target, such as the learned parameters, the concrete structure, etc. Compared with white-box attack, black-box attack has limited knowledge of the model but is closer to the practical scenarios. It is often modeled on querying the method by inputs, acquiring the final labels or confidence scores. Black-box attack roughly fails into transfer-based, score-based, and decision-based attack [9]. Transfer-based attack [25] utilizes the transferability of adversarial examples generated by white-box models. Score-based attack knows the predicted probability of classification, relying on approximated gradients to generate adversarial examples [15]. In decision-based attack, only the final label of classification is accessible [2] to the threat model. For black-box attack in visual object tracking, we assume that only the outputs of trackers (i.e., predicted bounding boxes) are available.

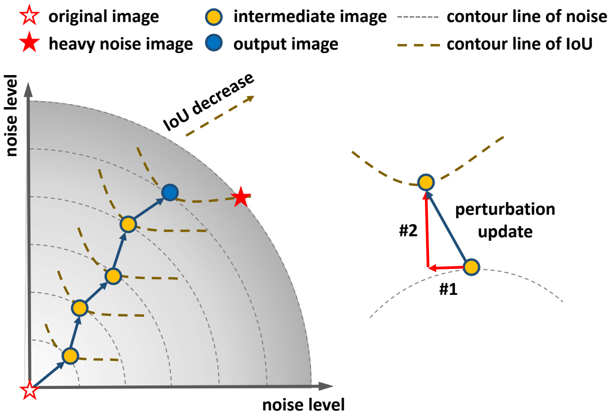
In the field of visual object tracking, some methods [3, 44, 16, 24, 14] explore the adversarial attack on different trackers. Chen et al. [3] propose a one-shot adversarial attack method by optimizing the batch confidence loss and the feature loss to modify the initial target patch. Yan et al. [44] design an adversarial loss to cool the hot regions on the heatmaps and shrink the predicted bounding box. All adversarial attack methods motioned above are summarized as white-box attack. In the real world, it is hard to know the concrete knowledge of trackers, causing the white-box attack methods less practical. In this work, we propose a novel decision-based black-box adversarial attack method for visual tracking. Motion consistency is taken into our attack method to further deteriorate the tracking performance.

3. Proposed Method

The proposed IoU attack aims to gradually decrease IoU scores (i.e., bbx overlap values) during iterations by using the minimal amount of noise. Figure 2 shows an intuitive view of the proposed IoU attack.

3.1. Motivation

Current studies on black-box attack mainly focus on static image recognition while the temporal motions of visual tracking are untouched. This is limited to attack deep trackers as the target object motion is maintained temporally. Meanwhile, deep trackers utilize temporally coherent schemes (i.e., search region constraint, and online update) to ensure tracking accuracy. The image content and temporal motions are equally important for black-box attack on visual tracking.



(a) Iterative perturbation update (b) Orthogonal composition

Figure 2. An intuitive view of IoU attack in the image space. In (a), we show that the increase of noise level positively correlates to the decrease of IoU scores but their directions are not exactly the same. The IoU attack method iteratively finds the intersection points (i.e., intermediate images) between each contour line of noise increase and IoU decrease. These intermediate images gradually decrease IoU scores with the lowest amount of noise. In (b), we show the orthogonal composition during each iteration. We generate noise hypothesis tangentially according to the current contour line (i.e., #1) and increase a small amount of noise in the normal direction (i.e., #2). The intersection point will be identified from the hypothesis that yields the lowest IoU at the same noise level. The updated perturbation in each iteration is the composition of #1 and #2.

The proposed IoU attack is to make the prediction results of one tracker deviate from its original performance. This is because of the tracking scenario where there is only one ground-truth bounding box (bbx) available (i.e., bbx annotation on the first frame). We define the original performance of one tracker is that it predicts one bbx on each frame without noise addition. By adding heavy-noisy perturbations, we make the same tracker predict another bbx and compute the spatial IoU score based on these two bbxs. Meanwhile, we use the bbx from the current frame and the one from the previous frame to compute a temporally coherent IoU score, which is then fused with the spatial IoU score. As state-of-the-art trackers demonstrate premier performance on the benchmarks, gradually decreasing the IoU scores by involving consecutive video frames indicates that their tracking performance deteriorates significantly. The IoU measurement suits different trackers as long as they predict one bbx for each frame.

3.2. IoU Attack

Figure 2 shows an intuitive view of how the proposed IoU attack gradually decreases the IoU scores between frames. Given a clean input frame, we first add heavy uniform noise on it to generate a heavy noise image where the IoU score is low. Along the direction from the clean image to the heavy noise image, the IoU scores gradually decrease while the noise level increases. The direction of IoU decrease positively correlates to that of noise increase but they are not exactly the same. IoU attack aims to progressively find a decreased IoU score while introducing the lowest amount of noise. The contour lines of IoU shown in Figure 2(a) indicate the tracker performance with regard to different noise perturbations, which can not be explicitly modeled in practice. From another perspective, IoU attack aims to identify one specific noise perturbation leading to the lowest IoU score among the same amount of noise levels. The identification process is fulfilled by orthogonal composition illustrated as follows.

We denote the original image on the t-th frame as I_0, the heavy noise image as H, and the intermediate image on the k-th iteration as I_k. In the (k+1)-th iteration, we first randomly generate several Gaussian distribution noise $\eta \sim \mathcal{N}(0,1)$ and select the tangential perturbation η from n of them as:

$$d(I_0, I_k) = d(I_0, I_k + \eta), \tag{1}$$

where d is the pixel-wise distance measurement between two images. Eq. 1 ensures tangential perturbations at the same noise level. The selected η^j ($j \in [1, 2, ..., n]$) is the perturbation tangential towards the contour line of noise level at the point I_k. We generate one I^j ($j \in [1, 2, ..., n]$) according to each η^j and use the tracker to predict a bbx B_t^j on it. Then, we define the IoU score S_{IoU} as:

$$S_{\text{IoU}} = \lambda \cdot S_{\text{spatial}} + (1 - \lambda) \cdot S_{\text{temporal}}, \tag{2}$$

$$S_{\text{spatial}} = \frac{B_t^{\text{orig}} \cap B_t^j}{B_t^{\text{orig}} \cup B_t^j}, \tag{3}$$

$$S_{\text{temporal}} = \frac{B_{t-1}^{\text{orig}} \cap B_t^j}{B_{t-1}^{\text{orig}} \cup B_t^j}, \tag{4}$$

where S_{spatial} denotes the spatial IoU score between the predicted bbx B_t^j and the original noise-free bbx B_t^{orig} at the t-th frame, S_{temporal} denotes the temporal IoU score with the original noise-free bbx B_{t-1}^{orig} at the (t-1)-th original frame, and λ is the scalar to balance the influence of spatial and temporal IoU scores. We attack S_{IoU} to perform both image content attack and temporal motion attack. In total, we obtain n IoU scores and select I^j whose S_{IoU} is lowest. An example of η^j is visualized as #1 in Figure 2(b).

Algorithm 1: Black-box IoU Attack

```
Input:  Input video $V$ with $M$ frames;
        Initialization perturbations $P_1 = 0$;
        Target bbx $B_1$ on the first frame;
Output: Adversarial examples of $M$ frames;
for $t = 2$ to $M$ do
    Get current frame $I_0$ and predict bbx $B_t^{\text{orig}}$ ;
    $I_0 = I_0 + \alpha \cdot P_{t-1}$;
    for $k = 0$ to $K$-1 do
        // Tangential direction
        Generate $N$ random perturbations $\eta$;
        Select $n$ of them according to Eq. 1;
        for $j = 1$ to $n$ do
            Predict the bbx $B_t^j$ on $I_k + \eta^j$ ;
            Compute $S_{\text{IoU}}$ according to Eq. 2;
        end
        Identify $j$ whose $S_{\text{IoU}}$ is lowest;
        // Normal direction
        Adjust $\epsilon$ to decrease $S_{\text{IoU}}$;
        Generate $I_{k+1}^j$ according to Eq. 5;
    end
    Obtain learned perturbations $P_t = I_{k+1}^j - I_0$;
    return $I_K^j$;
end
```

After getting the tangential perturbation, we denote η^j as neighboring hypothesis based on I_k and make $I_k + \eta^j$ towards the heavy noise image H as:

$$I_{k+1}^j = (I_k + \eta^j) + \epsilon \cdot \psi(H, I_k + \eta^j), \tag{5}$$

where ϵ controls the moving step towards H and $\epsilon \cdot \psi(H, I_k + \eta^j)$ is the perturbation following the noise increase direction (i.e., normal direction towards the contour line of noise level). We adjust the parameter ϵ moderately to limit the variation of perturbations. An example of $\epsilon \cdot \psi(H, I_k + \eta^j)$ is visualized as #2 in Figure 2(b). To this end, I_{k+1}^j is the intermediate image on the (k+1)-th iteration, consisting of the composed perturbations from both tangential and normal directions. We continuously perform the iteration until the IoU score is below the predefined threshold or the perturbations exceed the maximum. We transfer the learned perturbations P_t to the next few frames. The learned perturbations become the initialized perturbations, which are added on I_0 of (t+1)-th frame to encode temporal motion attack from previous frames. The pseudo code of the black-box IoU attack is shown in Algorithm 1.

3.3. Discussions and Visualizations

In this section, we visualize the variations of adversarial perturbations during IoU attack in Figure 3. Given an original image, we iteratively inject the adversarial perturbation as shown in the first row of Figure 3. With the increase of adversarial perturbations, the adversarial example drifts the

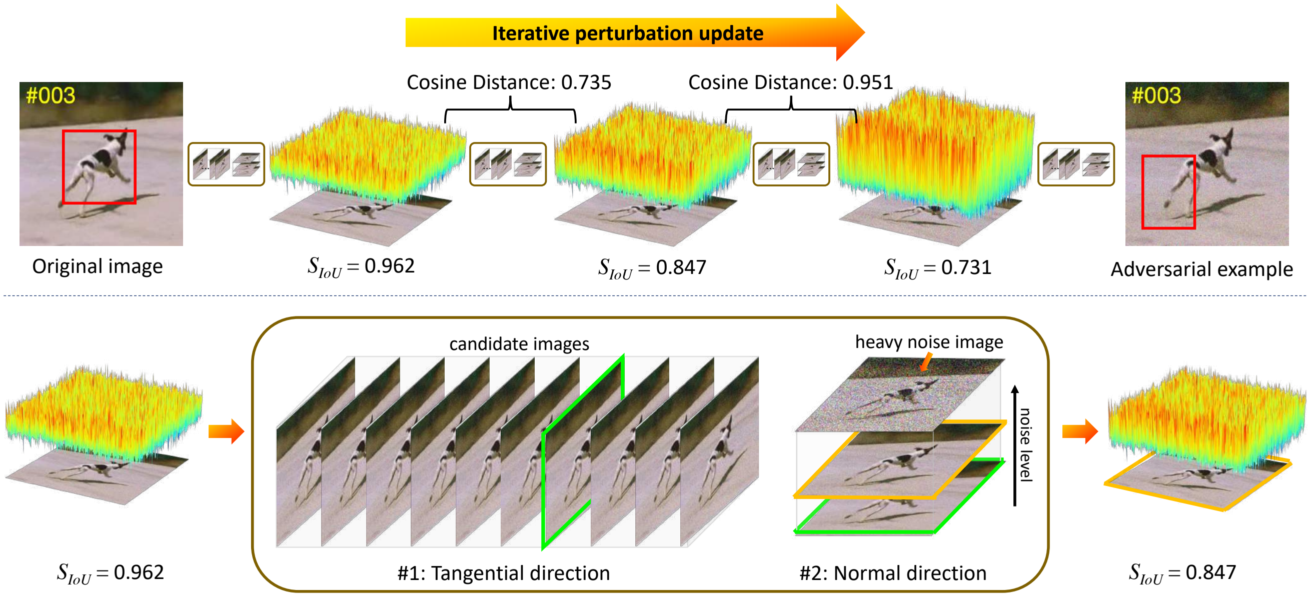


Figure 3. Variations of adversarial perturbations during IoU attack. The 3D response map above the image represents the difference between the original image and the adversarial example at different IoU scores. The IoU score decreases as the magnitude of perturbations increases. The variations of perturbations are illustrated in the first row. The orthogonal composition is shown in the second row, including tangential direction and normal direction. The image framed in green represents the minimal IoU score in the tangential direction and the image framed in yellow represents the moving step towards the heavy noise image.

Table 1. Comparison of tracking results with original sequences, random noise, and IoU attack of SiamRPN++ [22], DiMP [1] and LTMU [5] respectively on the VOT2019 [21] dataset.

Trackers	Accuracy ↑			Robustness ↓			Failures ↓			EAO ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.596	0.591	**0.575**	0.472	0.727	**1.575**	94	145	**314**	0.287	0.220	**0.124**
DiMP	0.568	0.567	**0.474**	0.277	0.373	**0.641**	55	74	**127**	0.332	0.284	**0.195**
LTMU	0.625	0.623	**0.576**	0.913	1.073	**1.470**	182	214	**293**	0.201	0.175	**0.150**

target from the original result and leads IoU scores to decrease. We compute the cosine distance between the perturbations from two consecutive intermediate images. The cosine distance indicates that the generated perturbations follow an increasing trend without fluctuation, decreasing the query numbers effectively in our black-box attack. During each iteration, we visualize the concrete orthogonal composition between the consecutive intermediate images for instance, as shown in the second row of Figure 3. We introduce several candidate images according to Eq. 1 and select the one with the minimal IoU score as the tangential direction (i.e., #1). Then, we move toward the heavy noise image in trails to make sure the IoU score decreases. We adjust the weight ϵ in Eq. 5 to constrain the variation of perturbation and output the result as the normal direction (i.e., #2). These two directions compose the orthogonal composition during each iteration. As a result, we hope the final perturbation preserves a lighter degree of noise than heavy random noise does, but the final perturbation can decrease the IoU scores heavily. In other words, our IoU attack makes larger degradation of IoU scores by injecting fewer perturbations.

4. Experiments

We validate the performance of our IoU attack on six challenging datasets, VOT2019 [21], VOT2018 [19], VOT2016 [20], OTB100 [42], NFS [18] and VOT2018-LT [19]. Detailed results are provided as follows.

4.1. Experiment Setup

Deployment of Trackers. In order to validate the generality of our black-box adversarial attack, we choose three representative trackers with different structures, SiamRPN++ [22], DiMP [1] and LTMU [5], respectively. SiamRPN++ is a typical detection based tracker with the siamese network. It compares the similarity between a target template and a search region with the region proposal network. The end-to-end learned tracker DiMP exploits both target and background appearance information to lo-

Table 2. Comparison of tracking results with original sequences, random noise, and IoU attack of SiamRPN++ [22], DiMP [1] and LTMU [5] respectively on the VOT2018 [19] dataset.

Trackers	Accuracy ↑			Robustness ↓			Failures ↓			EAO ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.602	0.587	**0.568**	0.239	0.365	**1.171**	51	78	**250**	0.413	0.301	**0.129**
DiMP	0.574	0.560	**0.507**	0.145	0.202	**0.400**	31	43	**85**	0.427	0.363	**0.248**
LTMU	0.624	0.622	**0.590**	0.702	0.805	**1.320**	150	172	**282**	0.195	0.178	**0.120**

Table 3. Comparison of tracking results with original sequences, random noise, and IoU attack of SiamRPN++ [22], DiMP [1] and LTMU [5] respectively on the VOT2016 [20] dataset.

Trackers	Accuracy ↑			Robustness ↓			Failures ↓			EAO ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.643	0.632	**0.605**	0.200	0.340	**0.802**	43	73	**172**	0.461	0.331	**0.183**
DiMP	0.599	0.592	**0.536**	0.140	0.168	**0.374**	30	36	**80**	0.449	0.404	**0.256**
LTMU	0.661	0.646	**0.604**	0.522	0.592	**0.904**	112	127	**194**	0.236	0.233	**0.170**

Table 4. Comparison of tracking results with original sequences, random noise, and IoU attack of SiamRPN++ [22], DiMP [1] and LTMU [5] respectively on the OTB100 [42] dataset.

Trackers	Success ↑			Precision ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.695	0.631	**0.499**	0.905	0.818	**0.644**
DiMP	0.671	0.659	**0.592**	0.869	0.860	**0.791**
LTMU	0.672	0.622	**0.517**	0.872	0.815	**0.712**

Table 5. Comparison of tracking results with original sequences, random noise, and IoU attack of SiamRPN++ [22], DiMP [1] and LTMU [5] respectively on the NFS30 [18] dataset.

Trackers	Success ↑			Precision ↑		
	Orig.	Rand.	Attack	Orig.	Rand.	Attack
SiamRPN++	0.509	0.466	**0.394**	0.601	0.550	**0.446**
DiMP	0.614	0.591	**0.545**	0.729	0.710	**0.658**
LTMU	0.631	0.579	**0.462**	0.764	0.699	**0.559**

cate the target precisely. LTMU is a long-term tracker, utilizing the meta-updater to update the tracker online for target prediction.

Implementation Details. We formulate the heavy noise image by injecting uniform noise into the clean image as feedback. The type of initial random noise at the same noise level is not sensitive to the degradation of tracking. We discontinue the iterative perturbation update when the IoU score is below the predefined score or the perturbations exceed the maximum. To sum up, the average query numbers of IoU attack are 21.2, 31.4 and 54.2 per frame for SiamRPN++, DiMP and LTMU, respectively.

4.2. Overall Attack Results

VOT2019. We implement the three trackers on the VOT2019 [21] dataset consisting of 60 challenging sequences. Different from other datasets, the VOT dataset has a reinitialization module. When the tracker loses the target (i.e., the overlap is zero between the predicted result and the annotation), the tracker will be reinitialized with the ground truth. Failures show the number of re-initialization. Accuracy evaluates the average overlap ratios of successfully tracking frames. Robustness measures the overall lost numbers. In addition, Expected Average Overlap (EAO) is evaluated by a combination of Accuracy and Robustness.

Table 1 shows the performance drops after IoU attack. We first test all trackers on original sequences. Then we implement our IoU attack method to generate the adversarial examples and evaluate the tracking results. SiamRPN++ leads to more failures than its original results, and the EAO score drops from 0.287 to 0.124. DiMP obtains a 16.5% drop on its accuracy score, which indicates our attack method leads to an obvious drift. The EAO score also drops dramatically from 0.332 to 0.195. Similarly, our IoU attack method reduces the EAO score of LTMU from 0.201 to 0.150. For further comparison, we also conduct experiments that inject the same level of random noise into the original sequences. Our generated perturbations decrease the IoU scores more dramatically than random noise.

VOT2018. There are 60 different sequences in the VOT2018 [19] dataset. All the trackers perform favorably on the original sequences. LTMU performs worse than the other two trackers since the re-detection module yields more reinitializations in VOT-toolkit. Table 2 shows that the performance of these trackers deteriorates obviously under IoU attack. Concretely, the accuracies of these three trackers get worse after the adversarial attack. These indicate that the trackers indeed deviate from their original results. The primary metric EAO scores are reduced by 68.8%, 41.9%, 38.5% for SiamRPN++, DiMP and LTMU, respectively.

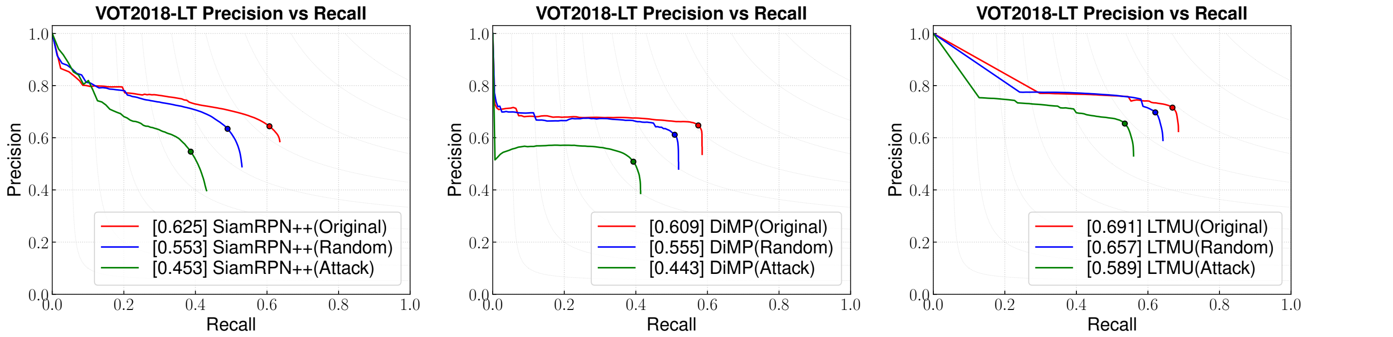


Figure 4. Precision and recall plots of IoU attack for SiamRPN++ [22], DiMP [1] and LTMU [5] respectively on the VOT2018-LT dataset [19]. We use *Attack* and *Random* to denote IoU attack and the same level of random noise. The legend is ranked by F-score.

VOT2016. Similarly, we also conduct the IoU attack method on the VOT2016 dataset [20], as shown in Table 3. These trackers perform much better than the above two datasets on the original sequences. However, IoU attack also reduces the EAO by 60.3%, 43.0%, 28.0% for SiamRPN++, DiMP and LTMU, respectively. Our IoU attack is more effective than the same level of random noise.

OTB100. The OTB100 [42] dataset includes 100 fully annotated video sequences. The evaluation has two main metrics, success and precision, by using the one-pass evaluation (OPE). We compare the results before and after IoU attack in Table 4. With IoU attack, the AUC scores of success significantly decline, accounting for 71.8%, 88.2% and 76.9% of original results for SiamRPN++, DiMP and LTMU, respectively. However, the AUC scores with random noise account for 90.8%, 98.2% and 92.6%, respectively.

NFS30. We also conduct IoU attack on the NFS30 [18] dataset consisting of 100 videos at 30 FPS with an average length of 479 frames. All sequences are manually labeled with nine attributes, like occlusion, fast motion, etc. And we adopt the same metrics used in the OTB100 dataset, as shown in Table 5. According to the AUC metric of success, SiamRPN++ obtains a 22.6% decrease after IoU attack while injecting the same level of random noise causes an 8.4% decrease. DiMP achieves an 11.2% decrease compared to a 3.7% decrease with random noise. LTMU gets a 26.8% decrease after IoU attack and an 8.2% decrease with random noise. IoU attack makes approximately triple drops compared to the same level of random noise.

VOT2018-LT. In order to further verify the effectiveness of our IoU attack, we conduct three trackers on a more challenging dataset VOT2018-LT [19]. It has 35 sequences with an average length of 4200 frames, which is much longer than other datasets and closer to practical applications. Each tracker needs to output a confidence score for the target being present and a predicted bounding box in each frame. Precision (P) and recall (R) are evaluated for a series of confidence thresholds, and the F-score is calculated as $F = 2P \cdot R/(P + R)$. The primary long-term tracking metric is the highest F-score among all thresholds. Figure 4 shows the results of precision and recall at different confidence thresholds before and after IoU attack. The results in the legend are ranked by F-score. The precision and recall both drop significantly after our IoU attack on three trackers. Our IoU attack method reduces the F-Score by 27.5%, 27.3% and 14.8% for SiamRPN++, DiMP and LTMU, respectively. All trackers after IoU attack perform poorly compared with injecting the same level of random noise. Our black-box attack method is proven to be also effective for long-term tracking.

Table 6. Ablation studies on IoU attack for SiamRPN++ [22], DiMP [1] and LTMU [5] on the VOT2018 [19] and VOT2016 [20] datasets. S_{temporal} represents the temporal IoU score and P_{t-1} represents the learned perturbation from historical frames.

Tracker	S_{temporal}	P_{t-1}	EAO ↑	
			VOT2018	VOT2016
SiamRPN++	No	Yes	0.149	0.189
	Yes	No	0.134	0.190
	Yes	Yes	**0.129**	**0.183**
DiMP	No	Yes	0.257	0.275
	Yes	No	0.261	0.295
	Yes	Yes	**0.248**	**0.256**
LTMU	No	Yes	0.147	0.184
	Yes	No	0.150	0.189
	Yes	Yes	**0.120**	**0.170**

4.3. Ablation Studies

To explore the temporal motion of visual object tracking in black-box attack, we separately compare the IoU attack method with or without involving temporal IoU scores in Eq. 2, as reported in Table 6. With the help of temporal IoU scores, the deep trackers get worse tracking accuracies than only using the spatial IoU scores on multiple datasets. In addition, we transfer the learned perturbation P_{t-1} into the

Table 7. Comparison with existing white-box and black-box attack methods for SiamRPN++ [22] with ResNet on the OTB100 [42] dataset.

Method	Succ. Drop	Prec. Drop	Type
CSA [44]	37.2	44.3	White-box
SPARK [14]	6.6	2.7	Black-box
UAP [26]	2.7	4.0	Black-box
Ours	**19.6**	**26.1**	Black-box

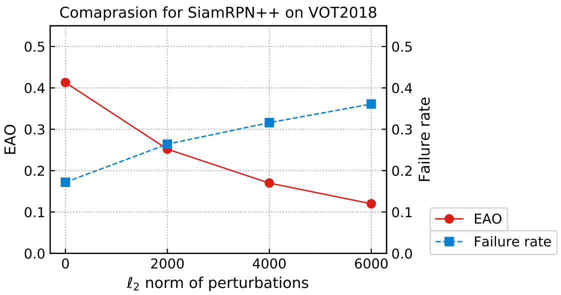


Figure 5. Comparison on EAO scores and failure rates of different perturbations for SiamRPN++ [22] on the VOT2018 [19] dataset.

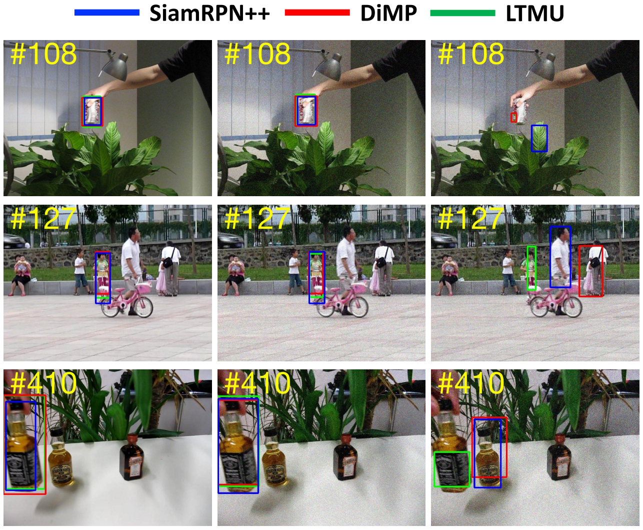
next few frames and use its weight to initialize the input for temporally consistent motion attack. We compare the IoU attack method with or without the transfer of historical perturbation P_{t-1} in Table 6. The overall performance metric EAO indicates that the transfer and initialization of previous perturbation indeed improve the attack effects and decrease the tracking accuracies. In addition, we also illustrate the attack performance with the variation of perturbations on the VOT2018 [19] dataset, as shown in Figure 5. The perturbations are measured by ℓ_2 norm. Failure rate represents the average rate of failure frames in the whole video. We observe that the attack performance gets worse with the increase of perturbations accordingly.

4.4. Comparison with Other Methods

Table 7 reports the comparison with existing white-box and black-box attack methods. Our black-box attack method without access to the network architecture of trackers performs slightly worse than the white-box attack method CSA [44] for tracking. SPARK [14] performs the transfer-based black-box attack and obtains a 6.6% success drop and a 2.7% precision drop. Our decision-based black-box attack significantly outperforms SPARK. In addition, we apply the perturbation from UAP [26] frame by frame, which is designed for attacking the classification in static images. Our method considering the temporal motion changes of the target objects achieves much greater success.

4.5. Qualitative Results

Figure 6 qualitatively shows the tracking results of our IoU attack for SiamRPN++ [22], DiMP [1] and LTMU [5] on three challenging sequences. We visualize the original tracking results in (a), the results with the same level of random noise in (b) and the results of our IoU attack in (c). In the original images, all these trackers locate the target objects and estimate the scale changes accurately. After generating the adversarial examples, these trackers estimate the target location inaccurately. However, the same level of random noise cannot drift the trackers, as shown in the second column. This indicates that the proposed IoU attack generates the optimized perturbations and maintains the same level of random noise.



(a) Original results (b) Random results (c) Attack results

Figure 6. Qualitative results of IoU attack on three challenging sequences from the OTB100 [42] dataset.

5. Concluding Remarks

In this paper, we propose an IoU attack method in the black-box setting to generate adversarial examples for visual object tracking. Without access to the network architecture of deep trackers, we iteratively adjust the direction of light-weight noise according to the predicted IoU scores of bounding boxes, which involve temporal motion in historical frames. Furthermore, we transfer the perturbations into the next frames to improve the effectiveness of attack. We apply the proposed method to three state-of-the-art representative trackers to illustrate the generality of our black-box adversarial attack for visual object tracking. The extensive experiments on standard benchmarks demonstrate the effectiveness of the proposed black-box IoU attack. We believe this work helps to evaluate the robustness of visual object tracking.

Acknowledgements. This work was supported by NSFC (61906119, U19B2035), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and Shanghai Pujiang Program.

References

[1] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019.

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *ICLR*, 2018.

[3] Xuesong Chen, Xiyu Yan, Feng Zheng, Yong Jiang, Shu-Tao Xia, Yong Zhao, and Rongrong Ji. One-shot adversarial attacks on visual tracking with dual attention. In *CVPR*, 2020.

[4] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, 2020.

[5] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In *CVPR*, 2020.

[6] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *CVPR*, 2017.

[7] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020.

[8] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, 2019.

[9] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, 2019.

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models. In *CVPR*, 2018.

[11] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *ICCV*, 2019.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[13] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*, 2020.

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

[15] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018.

[16] Shuai Jia, Chao Ma, Yibing Song, and Xiaokang Yang. Robust tracking against adversarial attacks. In *ECCV*, 2020.

[17] Ilchae Jung, Jeany Son, Mooyeol Baek, and Bohyung Han. Real-time mdnet. In *ECCV*, 2018.

[18] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, 2017.

[19] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, and et al. The sixth visual object tracking vot2018 challenge results. In *ECCV Workshop*, 2018.

[20] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Gustav Hager, Alan Lukezic, Abdelrahman Eldesokey, and et al. The visual object tracking vot2016 challenge results. In *ECCV Workshop*, 2016.

[21] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Cehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, et al. The seventh visual object tracking vot2019 challenge results. In *ECCV Workshop*, 2019.

[22] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

[23] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[24] Siyuan Liang, Xingxing Wei, Siyuan Yao, and Xiaochun Cao. Efficient adversarial attacks for visual object tracking. In *ECCV*, 2020.

[25] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017.

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[28] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016.

[29] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.

[30] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[31] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. *arXiv preprint arXiv:2103.05905*, 2021.

[32] Shi Pu, Yibing Song, Chao Ma, Honggang Zhang, and Ming-Hsuan Yang. Deep attentive tracking via reciprocative learning. In *NIPS*, 2018.

[33] Gege Qi, Lijun Gong, Yibing Song, Kai Ma, and Yefeng Zheng. Stabilized medical image attacks. *arXiv preprint arXiv:2103.05232*, 2021.

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

[35] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. CREST: Convolutional residual learning for visual tracking. In *ICCV*, 2017.

[36] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. VITAL: Visual tracking via adversarial learning. In *CVPR*, 2018.

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[38] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *CVPR*, 2020.

[39] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, 2019.

[40] Ning Wang, Wengang Zhou, Yibing Song, Chao Ma, Wei Liu, and Houqiang Li. Unsupervised deep representation learning for real-time tracking. *IJCV*, 2021.

[41] Rey Reza Wiyatno and Anqi Xu. Physical adversarial textures that fool visual object tracking. In *ICCV*, 2019.

[42] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 2015.

[43] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019.

[44] Bin Yan, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In *CVPR*, 2020.

[45] Bin Yan, Haojie Zhao, Dong Wang, Huchuan Lu, and Xiaoyun Yang. 'skimming-perusal'tracking: A framework for real-time and robust long-term tracking. In *ICCV*, 2019.

[46] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, 2019.

[47] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 2019.

[48] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *CVPR*, 2020.