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.

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...
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 exam-
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.

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 $\eta \sim N(0, 1)$ and select the tangential perturbation $\eta$ from $n$ of them as:

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

where $d$ is the pixel-wise distance measurement between two images. Eq. 1 ensures tangential perturbations at the same noise level. The selected $\eta^j (j \in [1, 2, ..., n])$ 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 $\eta^j$ and use the tracker to predict a bbx $B^j_t$ on it. Then, we define the IoU score $S_{IoU}$ as:

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

$$S_{spatial} = \frac{B^j_t \cap B^j_{t+1}}{B^j_t \cup B^j_{t+1}},$$  \hspace{1cm} (3)

$$S_{temporal} = \frac{B^j_{t+1} \cap B^{\omega}_{t+1}}{B^j_{t+1} \cup B^{\omega}_{t+1}},$$  \hspace{1cm} (4)

where $S_{spatial}$ denotes the spatial IoU score between the predicted bbx $B^j_t$ and the original noise-free bbx $B^\omega_t$ at the t-th frame, $S_{temporal}$ denotes the temporal IoU score with the original noise-free bbx $B^{\omega}_{t-1}$ at the (t-1)-th original frame, and $\lambda$ 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 $I^j$ is visualized as #1 in Figure 2(b).

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

$$I^j_{k+1} = (I_k + \eta^j) + \epsilon \cdot \psi(H, I_k + \eta^j),$$  \hspace{1cm} (5)

where $\epsilon$ controls the moving steps 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 $\epsilon$ 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, $I^j_{k+1}$ 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
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 $\epsilon$ 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>
<thead>
<tr>
<th>Trackers</th>
<th>Accuracy ↑</th>
<th>Robustness ↓</th>
<th>Failures ↓</th>
<th>EAO ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiamRPN++</td>
<td>0.596</td>
<td>0.591</td>
<td>0.575</td>
<td>0.472</td>
</tr>
<tr>
<td>DiMP</td>
<td>0.568</td>
<td>0.567</td>
<td>0.474</td>
<td>0.277</td>
</tr>
<tr>
<td>LTMU</td>
<td>0.625</td>
<td>0.623</td>
<td>0.576</td>
<td>0.913</td>
</tr>
</tbody>
</table>

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 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.

<table>
<thead>
<tr>
<th>Trackers</th>
<th>Accuracy ↑</th>
<th>Robustness ↓</th>
<th>Failures ↓</th>
<th>EAO ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiamRPN++</td>
<td>0.602 0.587 0.568</td>
<td>0.239 0.365 1.171</td>
<td>51 78 250</td>
<td>0.413 0.301 0.129</td>
</tr>
<tr>
<td>DiMP</td>
<td>0.574 0.560 0.507</td>
<td>0.145 0.202 0.400</td>
<td>31 43 85</td>
<td>0.427 0.363 0.248</td>
</tr>
<tr>
<td>LTMU</td>
<td>0.624 0.622 0.590</td>
<td>0.702 0.805 1.320</td>
<td>150 172 282</td>
<td>0.195 0.178 0.120</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Trackers</th>
<th>Accuracy ↑</th>
<th>Robustness ↓</th>
<th>Failures ↓</th>
<th>EAO ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiamRPN++</td>
<td>0.643 0.632 0.605</td>
<td>0.200 0.340 0.802</td>
<td>43 73 172</td>
<td>0.461 0.331 0.183</td>
</tr>
<tr>
<td>DiMP</td>
<td>0.599 0.592 0.536</td>
<td>0.140 0.168 0.374</td>
<td>30 36 80</td>
<td>0.449 0.404 0.256</td>
</tr>
<tr>
<td>LTMU</td>
<td>0.661 0.646 0.604</td>
<td>0.522 0.592 0.904</td>
<td>112 127 194</td>
<td>0.236 0.233 0.170</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Trackers</th>
<th>Success ↑</th>
<th>Precision ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orig. Rand. Attack</td>
<td>Orig. Rand. Attack</td>
</tr>
<tr>
<td>SiamRPN++</td>
<td>0.695 0.631 0.499</td>
<td>0.905 0.818 0.644</td>
</tr>
<tr>
<td>DiMP</td>
<td>0.671 0.659 0.592</td>
<td>0.869 0.860 0.791</td>
</tr>
<tr>
<td>LTMU</td>
<td>0.672 0.622 0.517</td>
<td>0.872 0.815 0.712</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Trackers</th>
<th>Success ↑</th>
<th>Precision ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orig. Rand. Attack</td>
<td>Orig. Rand. Attack</td>
</tr>
<tr>
<td>SiamRPN++</td>
<td>0.509 0.466 0.394</td>
<td>0.601 0.550 0.446</td>
</tr>
<tr>
<td>DiMP</td>
<td>0.614 0.591 0.545</td>
<td>0.729 0.710 0.658</td>
</tr>
<tr>
<td>LTMU</td>
<td>0.631 0.579 0.462</td>
<td>0.764 0.699 0.559</td>
</tr>
</tbody>
</table>

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 0.332 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.

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.

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.
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 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.

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

<table>
<thead>
<tr>
<th>Tracker</th>
<th>$S_{\text{temporal}}$</th>
<th>$P_{t-1}$</th>
<th>EAO $\uparrow$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>VOT2018</td>
</tr>
<tr>
<td>SiamRPN++</td>
<td>No</td>
<td>Yes</td>
<td>0.149</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>0.134</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td><strong>0.129</strong></td>
</tr>
<tr>
<td>DiMP</td>
<td>No</td>
<td>Yes</td>
<td>0.257</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>0.261</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td><strong>0.248</strong></td>
</tr>
<tr>
<td>LTMU</td>
<td>No</td>
<td>Yes</td>
<td>0.147</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>0.150</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td><strong>0.120</strong></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Succ. Drop</th>
<th>Prec. Drop</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSA [44]</td>
<td>37.2</td>
<td>44.3</td>
<td>White-box</td>
</tr>
<tr>
<td>SPARK [14]</td>
<td>6.6</td>
<td>2.7</td>
<td>Black-box</td>
</tr>
<tr>
<td>UAP [26]</td>
<td>2.7</td>
<td>4.0</td>
<td>Black-box</td>
</tr>
<tr>
<td>Ours</td>
<td>19.6</td>
<td>26.1</td>
<td>Black-box</td>
</tr>
</tbody>
</table>

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 $\ell_2$ 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.

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


