Spatial-aware Online Adversarial Perturbations Against Visual Object Tracking

Qing Guo\textsuperscript{1,2,}\textsuperscript{*}, Xiaoifei Xie\textsuperscript{2,}\textsuperscript{*}, Lei Ma\textsuperscript{3}, Zhongguo Li\textsuperscript{1}, Wei Feng\textsuperscript{1,}, Yang Liu\textsuperscript{2}

1 Tianjin University, China
2 Nanyang Technological University, Singapore
3 Kyushu University, Japan

Abstract

Adversarial attacks of deep neural networks have been intensively studied on image, audio, natural language, patch, and pixel classification tasks. Nevertheless, as a typical while important real-world application, the adversarial attacks of online video object tracking that traces an object’s moving trajectory instead of its category are rarely explored. In this paper, we identify a new task for the adversarial attack to visual object tracking: online generating imperceptible perturbations that mislead trackers along an incorrect (Untargeted Attack, UA) or specified trajectory (Targeted Attack, TA). To this end, we first propose a spatial-aware basic attack by adapting existing attack methods, i.e., FGSM, BIM, and C&W, and comprehensively analyze the attacking performance. We identify that online object tracking poses two new challenges: 1) it is difficult to generate imperceptible perturbations that can transfer across time/frames, and 2) real-time trackers require the attack to satisfy a certain level of efficiency. To address these challenges, we further propose the online incremental attack (OIA) that performs spatial-temporal sparse incremental perturbations online and makes the adversarial attack less perceptible. In addition, as an optimization-based method, OIA quickly converges to very small losses within several iterations by considering historical incremental perturbations, making it much more efficient than the basic attacks. The in-depth evaluation on the state-of-the-art trackers (i.e., SiamRPN with Alex, MobileNetv2, and ResNet-50) for OTB100 and VOT2018 demonstrates the effectiveness and transferability of OIA in misleading existing trackers under both UA and TA with minor perturbations.

1 Introduction

While deep learning achieves tremendous success over the past decade, the recently intensive investigation on image processing tasks e.g., image classification [Szegedy et al. 2013], Goodfellow, Shlens, and Szegedy 2014, Moosavi-Dezfooli, Fawzi, and Frossard 2016], object detection [Xie et al. 2017], and semantic segmentation [Metzen et al. 2017], reveal that the state-of-the-art deep neural networks (DNNs) are still vulnerable against adversarial examples. The minor perturbations on an image, although often imperceptible by human beings, can easily fool a DNN classifier, detector or segmentation analyzer, resulting in incorrect decisions. This draws great concerns especially when a DNN is applied in the safety- and security-critical scenarios. For a particular task, the domain-specific study and understanding of how adversarial attacks influence a DNN’s performance would be a key to reduce such impacts towards further robustness enhancement [Wei et al. 2018].

Besides image processing tasks, recent studies also emerge to investigate the adversarial attacks to other diverse types of tasks, e.g., speech recognition [Carlini and Wagner 2018, Qin et al. 2019, Cisse et al. 2017], natural language processing [Jin et al. 2019, Ren et al. 2019, Zhang et al. 2019], action recognition and object detection [Wei et al. 2018, Wei et al. 2019].

Visual object tracking (VOT), that performs online object localization and moving trajectory identification, is a typical while important component in many safety- and security-critical applications, with urgent industrial demands, e.g., autonomous driving, video surveillance, general-purpose cyber-physical system. For example, a VOT is often embedded into a self-driving car or unmanned aerial vehicle (UAV) as a key perception component, that drives the system to follow a target object.

Adversarial examples could mislead the car or UAV with incorrect perceptions, causing navigation into dangerous environments and even resulting in severe accidents. Therefore, it is of great importance to perform a comprehensive study of adversarial attacks on visual object tracking. Up to the present, however, there exist limited studies on the influence of the adversarial attack on VOT relevant tasks, without which the deployed real-world systems would be exposed to high potential safety risks.

Different from image, speech and natural language processing tasks, online object tracking poses several new challenges to the adversarial attack techniques. First, compared with existing sequential-input-relevant tasks, e.g., audios [Carlini and Wagner 2018], natural languages [Jin et al. 2019] or videos [Wei et al. 2018] for classification, that have access to the complete sequential data, object tracking processes incoming frames one by one in order. Under a current frame \( t \) under attack, all the previous frames (i.e.,
{1, 2, ..., i − 1} are already analyzed and cannot be changed. At the same time, the future frames (i.e., {i + 1, ...,}) are still unavailable and cannot be immediately attacked as well. With limited temporal data segments and the dynamic scene changes, it is even more difficult to generate imperceptible while effective adversarial perturbations that can transfer over time (i.e., multiple consecutive frames).

In addition, the object tracking often depends on a target designated object template cropped from the first frame of a video (Bertinetto et al. 2016; Li et al. 2018a) for further analysis. The different initially designated object might lead to different tracking analysis, which makes the universal adversarial perturbation (Moosavi-Dezfooli, Fawzi, and Frossard 2016) often ineffective.

Furthermore, online object tracking usually functions at real-time speed. Thus, it requires the attack techniques to be efficient enough so that the adversarial perturbation of the current frame can be completed before the next frame arrives. Although the gradient descent-based methods (e.g., FGSM (Goodfellow, Shlens, and Szegedy 2014), BIM (Kurakin, Goodfellow, and Bengio 2017)) are demonstrated to be effective in attacking the image classifier, they can still encounter efficiency issues in fooling the state-of-the-art trackers when multiple frames quickly arrive. It is also rather expensive for attacking on multiple frames in real-time, i.e., sparsity (Wei et al. 2018).

To better understand the challenges and uniqueness in attacking VOT, we first propose a spatial-aware basic attack method by adapting the existing state-of-the-art attacking techniques (i.e., FGSM, BIM, C&W) that are used to attack each frame individually. Our empirical study confirms that the basic attack is indeed ineffective for attacking VOT, due to the consecutive temporal frames in real-time. Based on this, we further propose the online incremental attack (OIA) method that can generate more imperceptible perturbations online in terms of both effectiveness and efficiency.

The main contributions of this paper are as follows:

- We formalize the the adversarial attack problem for VOT, i.e., generating imperceptible perturbations online to mislead visual trackers that traces an object, into an incorrect (Untargeted Attack, UA) or specified (Targeted Attack, TA) trajectory.

- We propose several basic attacks by adapting existing attacks (i.e., FGSM, BIM, C&W) and further perform an empirical study for better understanding the challenges of adversarial attacks on real-time object tracking.

- We propose a new online incremental attack (OIA) method that can efficiently generate more imperceptible perturbations for real-time VOT.

- In line with the basic methods, our in-depth evaluation demonstrates the effectiveness and efficiency of OIA in attacking the state-of-the-art SiamRPN trackers with Alex, MobileNetv2, and ResNet-50 models (Li et al. 2018a; Li et al. 2019) under both UA and TA. The generated attacks by OIA also exhibit strong transferability to the online updating variants of SiamRPN trackers.

2 Related Work

2.1 Adversarial examples

Extensive studies have shown the vulnerability of DNN against adversarial attacks (Ling et al. 2019; Akhtar and Mian 2018). Szegedy et al. (2013) initially pointed out the existence of adversarial attacks, and (Goodfellow, Shlens, and Szegedy 2014) proposed an efficient one-step method FGSM, that was later improved via iterative method (Kurakin, Goodfellow, and Bengio 2017) and momentum term (Dong et al. 2018). Similarly, Papernot et al. (2016) proposed the Jacobian-based Saliency Map Attack (JSMA) with high success rate and small perturbations, while (Carlini and Wagner 2017) realized effective attack by optimization methods (C&W) under different norms. Further adversarial perturbations were also extended to tasks like object detection (Xie et al. 2017; Li et al. 2018b; Zhao et al. 2019), and semantic segmentation (Xie et al. 2017; Moosavi-Dezfooli et al. 2017).

Recent works also confirmed the existence of adversarial examples in sequential data processing, e.g., speech recognition (Cisse et al. 2017; Carlini and Wagner 2018; Qin et al. 2019), and natural language (Gao et al. 2018; Jin et al. 2019) and video (Wei et al. 2018) processing. Different with these existing work, our attack aims to mislead visual tracker with limited online data access, i.e., the future frames are unavailable, the past frames cannot be attacked either. Among the most relevant work to ours, (Wei et al. 2018) proposed the $L_{2,1}$ norm-based attack to generate sparse perturbations for action recognition, under the condition that the whole video data is available and the perturbation of multiple frames can be jointly tuned. To further show the difference, we implement an attack for tracking with (Wei et al. 2018) and compare it with our method in the evaluation. (Li et al. 2018b) attacked the region proposal network (RPN) that is also used in the SiamRPN trackers (Li et al. 2018a). Nevertheless, this attack focuses on fooling image detectors to predict inaccurate bounding boxes, thus cannot be directly used to attack trackers that aim to mislead to an incorrect trajectory with online videos. (Wei et al. 2019) proposed the video object detection attack by addressing each frame independently, which is not suitable for online tracking where the tracker often runs at real-time speed. Another related work (Lin et al. 2017) studied when to attack an agent in the reinforcement learning context and used the degree of preference to action to decide the critical attacking time. In contrast, this work mainly explores how to use temporal constraints to online generate imperceptible and effective perturbations to mislead real-time trackers.

2.2 Visual object tracking

Visual object tracking is a fundamental problem in computer vision, which estimates the positions of a target object (specified at the first frame) over frames (Wu, Lim, and Yang 2015). The state-of-the-art trackers can be roughly summarized into three categories, including correlation filter-based (Danelljan et al. 2017; Lukežič et al. 2017), classification & updating-based (Nam and Han 2016; Song et al. 2018) and Siamese network-based (Bertinetto et al. 2016).
we crop an object template in the first frame. To locate the object at frame \( t \), we get object at frames. We denote the ground truth bounding box of the targeted object at the first frame, a tracker is to predict the bounding box \( b_t^{gt} \) such that \( \forall 1 \leq t \leq N, \text{IoU}(OT(X_t^{i}, T), b_t^{gt}) = 0 \), where \( \text{IoU}(\cdot) \) represents the Intersection over Union between two bounding boxes.

**Targeted Attack (TA).** Suppose a targeted trajectory \( \{p_{t}^{ta}\}_{t=1}^{N} \) desires the targeted tracking position at each frame. TA is to generate adversarial examples \( \{X_t^{i}\}_{t=1}^{N} \) such that \( \forall 1 \leq t \leq N, \text{ce}(OT(X_t^{i}, T)) = p_{t}^{ta} \), where \( \text{ce}(\cdot) \) shows the center position of the bounding box and \( p_{t}^{ta} \) depicts the targeted position at the \( t \)th frame.

Intuitively, UA is to make the SiamRPN-based trackers predict incorrect bounding boxes of a target object at all frames by adding small distortions to search regions while TA aims to intentionally drive trackers to output desired object positions specified by the targeted trajectory.

### 3.2 Basic Attack

We first propose the basic attacks at each frame by adapting existing adversarial methods, i.e., FGSM (Goodfellow, Shlens, and Szegedy 2014), BIM (Kurakin, Goodfellow, and Bengio 2017) and C&W (Carlini and Wagner 2017).

For UA, at frame \( t \), we formally define the problem of finding an adversarial example as follows:

\[
\min \mathcal{D}(X_t + E_t, T) \quad \text{such that} \quad OT(X_t + E_t, T) \neq b_t^{gt}
\]

where \( \mathcal{D} \) is the distance metric and \( E_t \) is the target distortion that changes the result of the tracker.

To achieve the goal, we define the objective function \( f \) such that \( OT(X_t + E_t, T) \neq b_t^{gt} \) if and only if \( f^{ua}(X_t + E_t, T) < 0 \):

\[
f^{ua}(X_t + E_t, T) = y_t^{gt} - \max_{i \neq gt, \text{IoU}(b_t^{(gt)}, b_t^{i}) = 0} (y_t^{i})
\]

where \( \{y_t^{i}, b_t^{i}\}_{i=1}^{N} = \phi_b(X_t + E_t, T), b_t^{gt} \) is the target object on \( X_t \) and \( y_t^{gt} \) is the activation value of \( b_t^{gt} \) at the frame \( t \).

For TA, at frame \( t \), we define the problem of finding a targeted adversarial example as follows:

\[
\min \mathcal{D}(X_t + E_t) \quad \text{such that} \quad \text{ce}(OT(X_t + E_t, T)) = p_t^{ta}
\]

where \( p_t^{ta} \) is the targeted position at the \( t \)th frame.

To achieve the goal, we define the objective function \( f \) such that \( OT(X_t + E_t, T) \neq b_t^{gt} \) if and only if \( f^{ta}(X_t + E_t, T) < 0 \):

\[
f^{ta}(X_t + E_t, T) = y_t^{gt} - \max_{\text{ce}(b_t^{i}) = p_t^{ta}} (y_t^{i})
\]

where \( \{y_t^{i}, b_t^{i}\}_{i=1}^{N} = \phi_b(X_t + E_t, T) \) and \( y_t^{gt} \) is the activation value of \( b_t^{gt} \) at the \( t \)th frame.

### 3.3 Empirical Study

Based on the basic attack that is conducted on single frame, we perform an empirical study on evaluating the effectiveness in attacking object tracking. In particular, we investigate two research questions: 1) how effective is the attack
by applying basic attack on each frame? 2) how is its impact of the temporal frames in the video?

To answer these questions, we perform two kinds of basic attacks on targeted attacks:

- **BA-E**: Online attacking each frame by using FGSM, BIM, and C&W to optimize Eq. \(2\), respectively.
- **BA-R**: Randomly select some frames and perform the basic attack on these frames using FGSM, BIM, and C&W. For frames between two selected frames, we use the perturbation from the first selected one to distort frames in the interval and see if the basic attacks could transfer across time. For example, we attack 1st and 10th frames with the basic attacks while distorting the 2th to 9th frames with the perturbation of 1st frame.

Note that BA-E is designed to answer the first question while BA-R is to answer the second question. To be specific, we configured two BA-R attacks. First, each frame is selected to be attacked with a probability 0.1 (denoted as BA-R1). Second, we perform the basic attack with an interval 10, i.e., attack at the 1th, 11th, 21th, . . . frame (denoted as BA-R2)

Table 1 shows the success rate, mean absolute perturbation, and average iteration per frame of BA-E, BA-R1, and BA-R2 for attacking SiamRPN-Alex-based tracker on OTB100 under TA. We see that: 1) BA-E methods via BIM and C&W get high success rate by attacking each frame. Nevertheless, their perturbations are large and attacking each frame with 10 iterations is time-consuming and beyond real-time tracker. Although FGSM is efficient, its success rate is much lower. 2) Randomly attacking 10% frames, i.e., BA-R1, is about 10 times faster than BA-E. However, the success rate drops significantly. 3) BA-R2 method attacking at every 10 frames is efficient while sacrificing the success rate. Compared with BA-R1, with the same attacking rate, i.e., 10% frames, BA-R2 has higher success rate than BA-R1. For example, base on BIM, BA-R2 has over two times larger success rate. It infers that perturbations of neighbor 10 frames have some transferability due to the temporal smoothness.

A case study based on BIM is shown in Fig. 1 where we use the three BA attacks to mislead the SiamRPN-Alex-based tracker to locate an interested object at the top left of the scene (targeted position in Fig. 1(c)). Instead of following the standard Siamese tracking pipeline, we...
crop the search region according to the ground truth and make sure the object are always at the center of a search region. We show the distance between the targeted position (Fig. 1(a) and tracking results, and the mean absolute perturbation (MAP) (Fig. 1(b)) at frame level. We reach consistent conclusion with Table 1. As the most simple solution, BA-E attacks the tracker successfully at some time (distance to the targeted position is less than 20) with the MAP around 5. However, the attack is inefficient and not suitable for real-time tracking. In addition, according to Fig. 1(c), the perturbations are large and perceptible. The results answer the first question: attacking on each frame is not effective, i.e., time-consuming and bigger MAP.

Consider the temporal property among the frames, if the attack can be transferred between the adjacent frames, we could only attack some frames while reducing the overhead, e.g., BA-R1 and BA-R2. Unfortunately, the results in Table 1 and Fig. 1 show that BA-R1 and BA-R2 only work at the specific frames, on which the attacks are performed.

The results answer the second question: the perturbations generated by BA is difficult to transfer to the next frames directly due to the dynamic scene in the video (see the results from BA-R1 and BA-R2).

### 3.4 Online Incremental Attack

Based on the empirical study results from basic attacks, we identify that attacking on each frame directly is not effective. As the frames are sequential and the nearby frames are very similar, our deep analysis found that transferability exists between nearby frames. Hence, we leverage the transferability and propose online incremental attack (OIA) that generates more imperceptible adversarial examples more efficiently for tracking. The intuition of OIA is that we still attack each frame, but apply previous perturbations on the new frame combined with small but effective incremental perturbation via optimization.

At frame $t$, the UA with OIA is formally defined as:

\[
\text{minimize } D(X_t, X_t + E_{t-1} + \epsilon_t)
\]

such that \( OT(X_t + E_{t-1} + \epsilon_t, T) \neq b^g_t \)

where \( E_{t-1} \) is the perturbation of the previous frame (i.e., \( t-1 \)th fame) and \( \epsilon_t \) is the incremental perturbation. Here, the ‘incremental’ means \( \epsilon_t = E_t - E_{t-1} \), and we further have \( E_t = \epsilon_t + \sum_{t_0}^{t-1} \epsilon_r \), where \( \{\epsilon_r\}_{t_0}^{t-1} \) are previous incremental perturbations, and \( \epsilon_{t_0} = E_{t_0} \). We denote \( t_0 \) as the start of an attack along the timeline. Based on Eq. 1 we introduce a new objective function by using \( L_{2,1} \) norm to regularize \( \{\epsilon_r\}_{t_0} \) that leads to small and spatial-temporal sparse \( \epsilon_t \).

\[
f^{\text{OA}}(X_t + \epsilon_t + \sum_{t_0}^{t-1} \epsilon_r, T) + \lambda \| \Gamma \|_{2,1}.
\]

where \( \Gamma = [\epsilon_{t_0}, ..., \epsilon_{t-1}, \epsilon] \) is a matrix that concatenates all incremental values.

Similarly, the TA with OIA is formally defined as follows:

\[
\text{minimize } D(X_t, X_t + E_{t-1} + \epsilon_t)
\]

such that \( ce(OT(X_t + E_{t-1} + \epsilon_t, T)) = p^*_r \).

---

**Algorithm 1:** Online adversarial perturbations for TA

**Input:** A video \( V = \{I_t\}_T \); the object template \( T \) specified at the first frame; targeted trajectory \( \{p^*_r\} \).

**Output:** Adversarial Perturbations \( \{E_t\}_T \).

Initialize the incremental perturbation set \( \mathcal{E} \) as empty:

\[
\text{for } t = 2 \text{ to } T \text{ do}
\]

\[
\text{Loading frame } I_t;
\]

\[
\text{Crop } X_t \text{ at the center of } b_t^g;
\]

\[
(y_t^r, b_t^g) = \arg\max_{y_t^r} \phi_t(X_t, T);
\]

\[
\text{if } \text{mod}(t, 30) = 0 \text{ then}
\]

\[
\text{max_iter} = 10;
\]

\[
\text{Empty } \mathcal{E};
\]

\[
t_0 = t;
\]

\[
\text{else}
\]

\[
\text{max_iter} = 2;
\]

\[
\epsilon_t = \text{OIA}(X_t, \mathcal{E}, T, \text{max_iter}); \text{ Add } \epsilon_t \text{ to } \mathcal{E} = \{\epsilon_r\}_{t_0}^{t-1};
\]

\[
E_t = \sum \mathcal{E};
\]

\[
(y_t^r, b_t^g) = \arg\max_{y_t^r} \phi(X_t + E_t, T);
\]

\[
t = t + 1;
\]

We also modify the objective function Eq. 2 by adding the \( L_{2,1} \) norm and obtain

\[
f^{\text{OA}}(X_t + \epsilon_t + \sum_{t_0}^{t-1} \epsilon_r, T) + \lambda \| \Gamma \|_{2,1}.
\]

When applying OIA on the SiamRPN-Alex-based trackers, we find following observations:

- **Spatial-temporal sparsity of incremental perturbations:** The incremental perturbations become gradually sparse along the space and time (see Fig. 1 (d)). This facilitates generating more imperceptible perturbations than BA methods. In addition, OIA gets the smallest MAP across all frames with higher success rate than BA-E on OTB100 (see Fig. 1 (b)).

- **Efficient optimization:** Fig. 1 (d) depicts the loss values during optimization from frame 41 to 49. At frame 41, it takes about 7 iterations to converge. However, at other frames, we obtain minimum loss in only two iterations. It enables more efficient attack than BA methods. As presented in Table 1, OIA only uses 2.25 iterations at average to achieve 78.9% success rate.

The sparsity and efficiency of OIA potentially avoid the high-cost iterations at each frame. In practice, we perform the OIA at every 30 frames and calculate \( E_{t_0} \) by optimizing Eq. 3 or Eq. 4 with 10 iterations, i.e., \( \{\epsilon_r\}_{t_0+30}^{t-1} \). We use the OIA to perform both UA and TA against visual object tracking and summarize the attack process of OIA for TA in Algorithm 1. At frame \( t \), we first crop a clean search region \( X_t \) and use it to get the object’s bounding box as \( b_t^g \). If \( t \) cannot be evenly divisible by 30, we optimize the objective function, i.e., Eq. 4 with 2 iterations and get \( \epsilon_t \). The, we add \( \epsilon_t \) into \( \mathcal{E} \) that stores previous incremental perturbations, i.e., \( \{\epsilon_r\}_{t_0}^{t-1} \), and obtain \( E_t = \sum \mathcal{E} \). If \( t \) can be evenly divisible by 30, we clear \( \mathcal{E} \) and start a new round attack by setting \( t_0 = t \).
4 Experimental Results

We design comprehensive experiments and leverage OIA to investigate the following research questions:

- **RQ1**: How effective is OIA for untargeted/targeted adversarial attacks compared with the existing techniques?
- **RQ2**: How is the transferability of OIA on other models?
- **RQ3**: How is the transferability of OIA on online updating trackers?

4.1 Setting

**Datasets.** We choose the widely used OTB100 (Wu, Lim, and Yang 2015), and VOT2018 (Kristan et al. 2018) as the subject datasets. OTB100 contains 100 videos with 59K frames and VOT2018 has 60 videos with about 21K frames.

**Models.** We select SiamRPN-based trackers (Li et al. 2018a, Li et al. 2019) with Alex, MobileNetv2, and ResNet-50, since these trackers are built on the same pipeline and achieve the state-of-the-art performance on various benchmarks. In particular, the models use AlexNet (Krizhevsky, Sutskever, and Hinton 2012), MobileNetv2 (Howard et al. 2017), and ResNet-50 (He et al. 2016) as backbones followed by a depth-wise correlation layer.

**Metrics.** We evaluate the effectiveness of adversarial perturbations on the basis of center location error (CLE) between predicted bounding boxes and the ground truth or targeted positions. We calculate precision drop for UA, success rate for TA, and MAP for both UA and TA:

- **Prec. Drop:** Following (Wei et al. 2019) and (Xie et al. 2017), for UA, we use precision drop of a tracker (after attacking) to evaluate the generated adversarial perturbations. The precision of a tracker is the rate of frames with CLE < 20 pixels between ground truth and predicted object positions among all frames.
- **Succ. Rate:** For TA, CLE is calculated between specified targeted positions and center positions of predicted object. Succ. Rate denotes the rate of frames (i.e., CLE < 20).
- **MAP:** Following (Wei et al. 2018), we use the mean absolute perturbation (MAP) to measure the distortion of adversarial perturbations. For a video dataset having \(D\) videos, we have \(\text{MAP} = \frac{1}{DKM} \sum_{d} \sum_{k} \sum_{c} \sum_{i} \sum_{c} \left| E_{d,k,c}(i,c) \right|\), where \(K\), \(M\), and \(C\) refer to the number of frames, pixels and channels, respectively.

**Configuration** Any gradient descent algorithm can be used to solve Eq. 5 and 6. Here, we use the sign gradient descent, with the step size of 1, followed by a clip operation. In Eq. 5 and 6, \(\lambda\) controls the regularization degree and we set it to a constant 0.00001.

For targeted attack, the targeted trajectory, i.e., \(\{p_{t}^{tr}\}_{t}^{T}\), is constructed by adding random offset values to the targeted position of previous frame, and we have \(p_{t}^{tr} = p_{t-1}^{tr} + \Delta p\), where \(\Delta p\) is within the range of 1 to 10. The generated trajectories are often more challenging than manual ones due to their irregular shapes.

4.2 Comparison Results (RQ1)

**Baselines.** Up to present, there still lacks research about adversarial attack on online object tracking. Therefore, we select and compare with the baselines by constructing the basic attacks and extending the existing video attack technique.

In Table 1, we have compared OIA with the Basic Attack methods, i.e., BA-E, BA-R1, and BA-R2. To further demonstrate the advantages of OIA over existing attack methods, we extend the BA-E such that it has the same configuration with OIA for a more fair comparison. To be specific, original BA-E attacks each frame with 10 iterations. However, in Algorithm 1, OIA attacks every 30 frames with 10 iterations while the frame in interval are attacked with only 2 iterations. We configure the new BA-E with the similar iteration configuration and adopt different optimization methods (i.e., FGSM, BIM (Kurakin, Goodfellow, and Bengio 2017), MI-FGSM (Dong et al. 2018), and C&W).

In addition, we also tried our best to compare with the existing technique, i.e., (Wei et al. 2018), which is designed for action recognition. However, it uses all frames of a video to predict the category and cannot directly be used for attacking online tracking. We made the extension of it, i.e., when we attack at frame \(t\), the previous 30 frames are used to generate the adversarial perturbations.

**Results.** Table 2 shows the targeted/untargeted attacking results on the two datasets. Column *Org. Prec.* gives the precision of the original tracker. Due to the large evaluation effort, we only perform the more comprehensive comparison on the smaller model, i.e., SiamRPN-Alex.

In Table 2, we see that OIA achieves high performance in attacking SiamRPN-Alex, MobileNetV2, and ResNet-50 with small perturbations. Compared with others, OIA reduces the most precision on the model SiamRPN-Alex for both of UA and TA. Consider the UA as an example, OIA achieves 78.9% precision drop on SiamRPN-Alex. For TA, OIA achieves 74.6% success rate, which is much better than the second best result (i.e., 41.8% from MI-FGSM). In addition, OIA generates imperceptible perturbations. For the results from (Wei et al. 2018), they generate the smallest imperceptible perturbations but the attacking is not effective.
Table 3: Transferability between subject models on OTB100. Values in UA and TA parts are Proc. Drop and Succ. Rate, respectively.

| Models          | Adversarial Perturbations | UA Attack | TA Attack |
|-----------------|---------------------------|-----------|-----------|
|                 |                           | Org. Prec. (%) | Proc. Drop (%) | MAP | Org. Prec. (%) | Proc. Drop (%) | MAP | Succ. Rate (%) | MAP |
| SiamRPN-Alex    | FGSM                      | 85.3       | 8.0        | 1.24 | 65.8          | 13.6          | 1.24 | 7.9           | 12.4 |
|                 | BIM                       | 85.3       | 72.1       | 2.17 | 65.8          | 57.4          | 2.28 | 35.8          | 2.14 |
|                 | MI-FGSM                   | 85.3       | 68.4       | 1.79 | 66.3          | 62.8          | 4.21 | 41.5          | 3.16 |
|                 | C&W                       | 85.3       | 91.2       | 1.37 | 65.8          | 50.6          | 1.26 | 55.7          | 1.27 |
|                 | Wei                       | 85.3       | 28.9       | 0.21 | 65.8          | 33.6          | 0.30 | 56.0          | 0.17 |
|                 | OIA                       | 85.3       | 48.9       | 2.10 | 65.8          | 41.8          | 2.10 | 45.4          | 1.16 |
| SiamRPN-Mob.    | OIA                       | 80.4       | 91.8       | 2.21 | 64.4          | 64.4          | 2.14 | 75.4          | 2.17 |
| SiamRPN-Res50   | OIA                       | 87.8       | 79.1       | 2.50 | 67.8          | 64.7          | 2.32 | 40.0          | 3.42 |

For example, the technique from Wei only achieves 25.9% precision drop in UA and 16.0% success rate in TA. Compared with others (i.e., FGSM, BIM, MI-FGSM, C&W), OIA generates the most imperceptible perturbations in UA, e.g., 1.04 on OTP 100. For TA, the perturbation of OIA is only larger than that of FGSM. However, FGSM achieves the worst attacking results, i.e., 7.9% and 4.3% success rate on OTB100 and VOT 2019. For other two larger models (i.e., SiamRPN-Mob. and SiamRPN-Res50), we can see that OIA is still useful to perform the targeted/untargeted attacks.

In summary, the results of Table 1 and Table 2 indicate the effectiveness of OIA in attacking the object tracking models with small distortions.

In addition to the quantitative analysis, we give a concrete example based on BIM and OIA (see Fig. 2). After OIA’s attacking, the SiamRPN-Alex tracker always produces bounding boxes on the targeted trajectory with a sparse perturbation, indicating the effectiveness of OIA. In contrast, the tracker can still predict bounding boxes that tightly wrap the object on adversarial examples generated by BIM.

4.3 Transferability Across Models (RQ2)

In this section, we discuss the transferability across models, which is to apply perturbations generated from one model to another. In Table 3, the values in the UA part are the Prec. Drop while the values for TA correspond to the Succ. Rate. We see that the transferability across models also exists in attacking object tracking. All attack methods lead to the precision drop to some extent. For example, the perturbations generated by SiamRPN-Res50 cause the precision of SiamRPN-Mob. drop 16.1, which is a large performance degradation in tracking evaluation. For TA, after transferability, the success rate is around 6.5 for all cases. Such limited transferability may be caused by the insufficient iterations during online process and can be further studied in the future. These results answer the second question that the transferability of an attack across subject models do exist in attacking visual object tracking.

Table 4: Transferability of SiamRPN-Alex, MobileNetv2, and ResNet-50 to online updating trackers on OTB100.

| Attacks | UA Attack | TA Attack |
|---------|-----------|-----------|
|         | Org. Prec. (%) | Proc. Drop (%) | MAP | Org. Prec. (%) | Proc. Drop (%) | MAP | Succ. Rate (%) | MAP |
| DSiamRPN-Alex | 86.6       | 78.8       | 1.05 | 65.9          | 3.19          |
| DSiamRPN-Mob.  | 87.8       | 91.1       | 2.19 | 65.2          | 3.19          |
| DSiamRPN-Res50 | 90.1       | 75.4       | 2.52 | 35.6          | 3.41          |

4.4 Transferability to Online Updating Trackers (RQ3)

In tracking area, it has been known that online updating module helps trackers to adapt to target and background changes. Thus, it is interesting to explore how adversarial perturbations would affect the online updating trackers. To this end, we construct three online updating trackers with dynamic Siamese tracking (DSiam) (Guo et al. 2017) based on the three models, and obtain three trackers: DSiamRPN-Alex, MobileNetv2, and ResNet-50. We then use the adversarial perturbations generated from SiamRPN-Alex, MobileNetv2, and ResNet-50 to attack their online updating versions respectively.

From Table 4, we observe that: 1) DSiam indeed improves the precision of SiamRPN-Alex, MobileNetv2, and ResNet-50 according to the results in Table 2. 2) The adversarial perturbations from offline models is still effective for their online versions with the precision drops being 78.5%, 81.1%, and 72.0% which are similar but slightly smaller than the results in Table 2. These results answer the third research question: with OIA, the adversarial attacks generated on offline models can be transferred to the online updating versions, effectively.

5 Conclusion

In this paper, we explored adversarial perturbations for misleading the online visual object tracking along an incorrect (untargeted attack, UA) or specified (targeted attack, TA) trajectory. An optimization-based method, namely online incremental attack, was proposed to overcome the challenges introduced in this new task. OIA optimizes incremental perturbations with an $L_{2,1}$ regularization norm and considers the influence of historical attacking results, thus is more effective. Experimental results on OTB100 and VOT2018 showed that OIA successfully fool state-of-the-art trackers. Moreover, we found that adversarial perturbations have good transferability to their online updating versions.
[Wiyatno and Xu 2019] Wiyatno, R. R., and Xu, A. 2019. Physical adversarial textures that fool visual object tracking. In arXiv:1904.11042.

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

[Xie et al. 2017] Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. L. 2017. Adversarial examples for semantic segmentation and object detection. In ICCV, 1378–1387.

[Zhang et al. 2019] Zhang, H.; Zhou, H.; Miao, N.; and Li, L. 2019. Generating fluent adversarial examples for natural languages. In ACL, 5564–5569.

[Zhao et al. 2019] Zhao, Y.; Zhu, H.; Liang, R.; Shen, Q.; Zhang, S.; and Chen, K. 2019. Seeing isn’t believing: Practical adversarial attack against object detectors. In CCS.

[Zhu et al. 2018] Zhu, Z.; Wang, Q.; Li, B.; Wei, W.; Yan, J.; and Hu, W. 2018. Distractor-aware siamese networks for visual object tracking. In ECCV.