MDPI

Article

Security in Transformer Visual Trackers: A Case Study on the Adversarial Robustness of Two Models

Peng Ye [1,2,3], Yuanfang Chen [1,2]*0000-0002-1557-087X, Sihang Ma [1,2], Feng Xue [4], Noel Crespi [5]0000-0003-2962-192X, Xiaohan Chen [1,2] and Xing Fang [1,2]

[1] School of Cyberspace, Hangzhou Dianzi University, China; yuanfang.chen.tina@gmail.com
[2] Key Laboratory of Discrete Industrial Internet of Things of Zhejiang, China
[3] DBAPPSecurity Co., Ltd.
[4] ZJU-Hangzhou Global Scientific and Technological Innovation Center, Zhejiang University, China; Henryxue@zju.edu.cn
[5] Institut Polytechnique de Paris, France; noel.crespi@mines-telecom.fr
* Correspondence: yuanfang.chen.tina@gmail.com; Tel.: +86-137-3545-0984

Abstract: Visual object tracking is an important technology in camera based sensor networks, which has a wide range of practicability in auto drive system. A transformer is a deep learning model that adopts the mechanism of self-attention, and it differentially weights the significance of each part of the input data. It has been widely applied in the field of visual tracking. Unfortunately, the security of the transformer model is unclear. It makes such transformer-based applications be exposed to security threats. In this work, the security of the transformer model is investigated with the important component of autonomous driving, visual tracking. Such deep-learning-based visual tracking is vulnerable to adversarial attacks, so adversarial attacks are implemented as the security threats to conduct the investigation. First, adversarial examples are generated on top of video sequences to degrade tracking performance, and the frame-by-frame temporal motion is taken into consideration when generating perturbations over the predicted tracking results. Then, the influence of perturbations on performance is sequentially investigated and analyzed. Finally, numerous experiments on OTB100, VOT2018, and GOT-10k data sets demonstrate that the executed adversarial examples are effective on the performance drops of the transformer-based visual tracking.

Keywords: autonomous driving; visual tracking; adversarial attacks; transformer model

Citation: Peng Ye; Yuanfang Chen; Sihang Ma; Feng Xue; Noel Crespi; Xiaohan Chen and Xing Fang. Security in Transformer Visual Trackers: A Case Study on the Adversarial Robustness of Two Models. *Journal Not Specified* **2024**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

0. Introduction

In recent years, autonomous vehicles have relied heavily on advanced sensor technologies, such as LIDAR, radar, GPS, and ultrasonic sensors, to navigate and understand their environments. Cameras, as a significant part of this sensor suite, provide crucial visual data for tasks like target tracking, traffic sign recognition, and lane detection. This image data plays a pivotal role in understanding dynamic scenes and tracking moving objects for safe autonomous driving. However, reliance on image data also brings particular vulnerabilities. Visual target tracking, which primarily depends on this camera-based image data, has seen remarkable improvements with the advent of deep learning models, particularly transformer. The transformer is a foundation model to drive a paradigm shift in artificial intelligence, and it has attracted increasing attention due to its remarkable ability to capture long-range dependencies and model sequential data, and it learns context and thus meaning by tracking relationships in sequential data. In transformer models, the self-attention technique is applied to detect subtle ways even distant data elements in a series influence and depend on each other. On these capabilities basis, the transformer model is driving a wave of advances in machine learning, and it greatly improves the performance in visual tracking. However, the security of the transformer model in visual tracking has not been thoroughly investigated yet. Although the transformer model has shown impressive performance in many tasks, it is vulnerable to adversarial attacks which can cause the model to

produce incorrect outputs or even fail completely. In visual tracking, adversarial examples have successfully attacked deep learning tasks such as image classification, object detection, and semantic segmentation. Adversarial attacks are particularly concerning in the context of transformer-model-based visual tracking, where the consequences of a misclassification or a false positive can be severe. The adversarial example study in deep visual tracking can not only help people understand its weakness, but also direct to improve the robustness of algorithms in visual tasks. Therefore, it is important to investigate the robustness and the security of deep-learning-based trackers.

In this paper, our work aims to investigate the security of transformer models in visual tracking, and evaluate their robustness against different types of adversarial attacks. Specifically, the vulnerability of the transformer models in visual tracking is explored to white-box [1], [2], [3], gray-box [4], [5], [6], [7] and black-box attacks [8], [9]. Moreover, this paper analyzes the impact of different attack methods on the tracking performance. The goal of our work is to provide insights into the security of the transformer models in visual tracking, and identify potential vulnerabilities that need to be addressed in future research.

Three attacks are deployed in the investigation experiments, cooling-shrinking attack [4], IoU attack [8] and RTAA attack [1], and the experiments are carried on three data sets, OTB100 [10], VOT2018 [11] and GOT-10k [12].

Figure 1 gives an example: the RTAA attack causes two transformer-model-based trackers to track wrong targets.

(a) Original Tracking Results (b) Adversarial Attack Results

Figure 1. The adversarial attack, RTAA, in two transformer-model-based trackers (TransT [13] and MixFormer [14]). The TransT tracker effectively locates targets in the original video sequences. The MixFormer utilizes the flexibility of attention operations, and there is a mixed attention module for simultaneous feature extraction and target information integration. The adversarial attack strategy decreases the tracking accuracy as shown in (b), with the RTAA attack, i.e., the TransT and MixFormer trackers output incorrect bounding boxes to track wrong targets.

The contributions of this paper are summarized as follows:

1. Investigation and analysis. Adversarial attacks against visual tracking tasks are investigated to analyze the tracking principle, the advantage and weakness of the transformer-model-based trackers. Moreover, the influence of the adversarial attacks

is studied as well. It is important to direct the design of robust and secure deep-learning-based trackers for visual tracking.

2. Implementation and verification. Three adversarial attacks are implemented to perform the attacks on the transformer-model-based visual tracking, and the effectiveness of these attacks is verified on three data sets.

1. Adversarial Attacks on Transformer-based Visual Tracking

1.1. Transformer Architecture

The transformer model is introduced by Vaswani *et al.*, and applied in machine translation. It is an architecture for transforming one sequence into another with the help of attention-based encoders and decoders. The attention mechanism takes an input sequence into each step, and decides at each step, to facilitate capturing the global information from the input sequence.

The transformer architecture has been used to replace recurrent neural networks in these sequential tasks: natural language processing, speech processing and computer vision, and gradually extended to handle non-sequential problems.

As the important mechanism of transformer architecture, the attention mechanism has been introduced into the tracking field. In [15], Choi *et al.* adopt channel-wise attention to provide the matching network with target-specific information. It merely borrows the concept of attention to conduct model or feature selection. In [16], Yu *et al.* explore both self-attention and cross-branch-attention to improve the discriminative ability of target features before applying the depth-wise cross correlation. In [17], Du *et al.* propose CGACD to learn attention from the correlation result of the template and search region, and then adopt the learned attention to enhance the search region features for further classification and regression. These works have improved the tracking accuracy with the attention mechanism, but they still highly rely on the correlation operation in fusing the template and search region features. In [13], Chen *et al.* design an attention-based network to directly fuse template and search region features without using any correlation operation.

1.2. Transformer Tracking

Transformer tracking is a state-of-the-art object tracking method which uses the transformer model to achieve accurate and robust object tracking. Compared to traditional object tracking methods, transformer tracking has shown superior performance in handling object deformation and occlusion. The key idea of transformer tracking is to represent each object as a vector learned by the transformer model. During tracking, the feature representation of the object is first converted into a vector and fed into the transformer model for processing, which generates a new representation of the object. The location and state of the object in the next frame are predicted based on the similarity between the old and new representations. Transformer tracking has several advantages over traditional tracking methods. First, the transformer model is capable of capturing the context information of the object, making the tracker more robust to object deformation and occlusion. Second, the representation vector of the object can be adapted dynamically during tracking, which allows the tracker to better adapt to the object's motion and deformation. Finally, pre-training can be applied to the transformer model to accelerate training and improve the tracking performance.

There have been several recent studies on transformer tracking, such as TransT [13], TMT [18], STARK [19], AiATrack [20], OSTrack [21], SwinTrack [22], TFITrack [23] and TrTr [24]. They utilize the encoder-decoder network to extract the global and rich contextual inter-dependencies. In addition, MixFormer is presented in [14] as a compact tracking framework, and it is built upon transformers. It is proposed to simplify the multi-stage pipeline of feature extraction, target information integration, and bounding box estimation. Moreover, it unifies the process of feature extraction and target information integration.

1.3. Adversarial Attacks on Transformer Tracking

Vision is the core and foundation of tracking, and the adversarial robustness of the vision transformer decides the robustness of the tracking with the transformer framework. In recent years, the vision transformer has achieved attention. In [25], the authors have showed that standard vision transformer models perform more robust than standard CNNs under adversarial attacks. In [26], the authors have revealed that vision transformer models are not more robust than CNNs, if both are trained in the same training framework. It is observed that the accuracy of standard models can be easily reduced to near zero under standard attacks. In addition, Fu *et al.* [27] studied attacking vision transformer models in a patch-wise approach, and it reveals the unique vulnerability of vision transformer models. To boost the adversarial robustness of vision transformer models, in [28], authors have explored multiple-step adversarial training to the vision transformer models. However, multi-step adversarial training is computationally expensive. To reduce computational cost, in [29], Wu *et al.* took the step of exploring fast single-step adversarial training on vision transformer models.

2. Generating Adversarial Examples

Three attack methods are implemented, they are: cooling-shrinking attack [4], IoU attack [8] and RTAA attack [1], and they are acted on TransT [13] and MixFormer [14].

2.1. Attack Principles

The attack principles of three attack methods are analyzed in detail as follows.

Cooling-shrinking attack's principle. In the cooling-shrinking attack, the proposed adversarial perturbation generator aims to deceive the SiamRPN++ tracker by making the target invisible and leading to tracking drift. This is achieved by training the generator with a cooling-shrinking loss. The generator is designed to attack either the search regions or the template, where the search regions are the target located, and the template is given in the initial frame.

The designed cooling-shrinking loss is composed of the cooling loss L_C to interfere the heat maps M_H, and the shrinking loss L_S to interfere the regression maps M_R, where the heat maps M_H and the regression maps M_R are important components of the SiamRPN++ tracker.

In the generator, the cooling loss L_C is designed to cool down the hot regions where the target may exist on, causing the tracker to lose the target, and the shrinking loss L_S is designed to force the predicted bounding box to shrink, leading to error accumulation and tracking failure.

IoU attack's principle. The IoU attack method aims to decrease the IoU scores between the predicted bounding boxes and ground truth bounding boxes in a video sequence, indicating the degradation of tracking performance. It is designed to counter existing black-box adversarial attacks that target static images for image classification. Unlike the existing black-box adversarial attacks, the IoU attack generates perturbations by considering predicted IoU scores from both current and previous frames. By decreasing the IoU scores, the IoU attack reduces the frame-by-frame accuracy of coherent bounding boxes in video streams. During the IoU attack, learned perturbations are utilized and transferred to subsequent frames to initiate a temporal motion attack. In the IoU attack, there is an increase in noise level as the IoU scores decrease, but this relationship is not linear: in an IoU attack, a clean input frame is subjected to the addition of heavy uniform noise, resulting in a heavily-noised image with a low IoU score. During the addition process, the IoU scores gradually decline as the noise level increases.

The following employed strategy achieves the effectiveness and imperceptibility of the IoU attack in video streams: there exists a positive correlation between the direction of decrease in IoU and the direction of increase in noise. However, this relationship is not linear. The IoU attack gradually reduces the IoU score for each frame in a video stream by adding the minimum amount of noise. It identifies the specific noise perturbation that

results in the lowest IoU score among an equal amount of noise levels through orthogonal composition.

RTAA attack's principle. The RTAA attack takes temporal motion into consideration over the estimated tracking results frame-by-frame.

The RTAA attack creates a pseudo classification label and a pseudo regression label, and both labels are used to design the adversarial loss. The adversarial loss is set to make L_c and L_r be the same when correct and pseudo labels are used separately, where the L_c denotes the binary classification loss, and the L_r is the bounding box regression loss, and they are two important parameters in deep visual tracking algorithms.

In deep visual tracking, the binary classification loss is a measure used to evaluate the performance of a visual tracking algorithm. Visual tracking is often framed as a binary classification problem, where the goal is to distinguish between the target and the background. The binary classification loss function in visual tracking measures the difference between the predicted class probabilities and the true class labels. In this case, the two classes are the target and the background. The loss function is used to train the visual tracking algorithm and adjust its parameters so that it improves its ability to accurately track the target over time. Moreover, the bounding box regression loss in visual tracking is a measure used to evaluate the performance of a visual tracking algorithm in predicting the location and the size of the bounding box that encloses the target. In visual tracking, the goal is to track the target of interest over time, and the bounding box regression loss function is used to adjust the parameters of the tracking algorithm so that it can accurately predict the location and size of the bounding box that encloses the target in each frame of the video sequence.

2.2. Advantages and Weaknesses of Attacks

Cooling-shrinking attack's advantages. There are two advantages: (i) the use of a cooling-shrinking loss allows for fine-tuning of the generator to generate imperceptible perturbations while still effectively deceiving the tracker, and (ii) the method is able to attack the SiamRPN++ tracker, which is currently one of the most powerful trackers, achieving the state-of-the-art performance on almost all tracking data sets.

Cooling-shrinking attack's weaknesses. There are three weaknesses: (i) the method is specifically designed to attack the SiamRPN++ tracker, and may not be effective against other types of trackers, and (ii) the generator is trained with a fixed threshold, so it may not be effective against different scenarios or environments, and (iii) the attack method may have limited use in real-world applications, as adding adversarial perturbations to targets being tracked.

IoU attack's advantages. There are three advantages: (i) the IoU attack involves both spatial and temporal aspects of target motion, making it more comprehensive and challenging for visual tracking, (ii) the method uses a minimal amount of noise to gradually decrease the IoU scores, making it more effective in terms of computational costs, and (iii) the IoU attack can be applied to different trackers as long as they predict one bounding box for each frame, making it more versatile.

IoU attack's weaknesses. There are three weaknesses: (i) the exact relationship between the noise level and the decrease of IoU scores is not explicitly modeled, making it difficult to optimize the noise perturbations, (ii) the method involves a significant amount of computation during each iteration, which might affect its efficiency in real-world applications, and (iii) the method relies on the assumption that the trackers use a single bounding box prediction for each frame, which might not always be the case in some complex scenarios.

RTAA attack's advantages. There are three advantages: (i) the RTAA attack generates adversarial perturbations based on the input frame and the output response of deep trackers, which makes the adversarial examples more effective and realistic, (ii) the attack uses the tracking-by-detection framework, which is widely used in computer vision tasks and helps to increase the robustness of the attack, and (iii) the method can effectively

confuse the classification and regression branches of the deep tracker, which results in rapid degradation in performance.

RTAA attack's weaknesses. There are four weaknesses: (i) the method relies on a fixed weight parameter λ, which may not be optimal for different types of deep trackers and attack scenarios, (ii) the method uses a random offset and scale variation for the pseudo regression label, which may not be effective for all tracking scenarios, (iii) the method requires multiple iterations to produce the final adversarial perturbations, which increases the computational complexity of the attack, and (iv) the method considers the adversarial attacks in the spatiotemporal domain, which may limit its applicability to other computer vision tasks that do not have a temporal aspect.

2.3. Transformer Tracking Principles

TransT's principle. Correlation acts as an important role in tracking. However, the correlation operation is a local linear matching process, which easily leads to lose semantic information and falls into local optimum. To address this issue, inspired by transformer architecture, TransT [13] is proposed with the attention-based feature fusion network, and it combines the template and search region features solely using an attention-based fusion mechanism.

TransT consists of three components: backbone network, feature fusion network and prediction head. The backbone network extracts the features of the template and the search region, separately. With the extracted features, then, the features are enhanced and fused by the proposed feature fusion network. Finally, the prediction head performs the binary classification and bounding box regression on the enhanced features to generate the tracking results.

MixFormer's principle. To simplify the multi-stage pipeline of tracking, and unify the process of feature extraction and target information integration, a compact tracking framework is proposed in [14], termed as MixFormer, which is built upon transformers.

MixFormer utilizes the flexibility of attention operations, and uses a mixed attention module, for simultaneous feature extraction and target information integration. This synchronous modeling scheme allows to extract target-specific discriminative features, and performs the extensive communication between the target and search areas. MixFormer simplifies the tracking framework by stacking multiple mixed attention modules, with embedding progressive patches and placing a localization head on top. In addition, to handle multiple target templates during online tracking, an asymmetric attention scheme is designed in the mixed attention module, to reduce computational cost, and an effective score prediction module is proposed to select high-quality templates.

2.4. Investigation Experiments and Analyses

Investigation experiments evaluate the robustness of tracker models based on the transformer framework, namely Transformer and MixFormer, against three distinct adversarial attack methods, and the evaluation is performed on three foundational benchmark datasets: OTB2015 [10], VOT2018 [11], and GOT-10k [12]. The investigated attack methods encompass white-box attack (RTAA attack), semi-white-box attack (CSA attack), and black-box attack (IoU attack). The objective is to comprehensively assess the vulnerability of these trackers under varying degrees of adversarial perturbations, shedding light on their limitations and potential defense strategies. The findings from this study contribute to enhancing the overall reliability and security of transformer-based trackers in real-world scenarios.

Standard evaluation methodologies are adopted on the benchmark datasets. For the OTB2015 [10] dataset, the one-pass evaluation (OPE) is utilized, which employs two key metrics: precision curve and success curve. The precision curve quantifies the center location error between the tracked results and the ground truth annotations, computed using a threshold distance, such as 20 pixels. The success curve measures the overlap ratio

between the detected bounding boxes and the ground truth annotations, reflecting the accuracy of the tracker at different scales.

This study evaluates object tracking algorithms on the VOT2018 [11] dataset using accuracy, robustness, failures, and expected average overlap (EAO) as evaluation metrics. Accuracy measures the precision of tracking algorithms in predicting the target's position, while robustness assesses the algorithm's resistance to external disturbances. Failures count the number of times the tracking process fails, and expected average overlap provides a comprehensive metric considering both accuracy and robustness, calculated by integrating the success rate curve to evaluate the overall performance of the object tracking algorithms.

The average overlap (AO) and success rate (SR) are adopted as evaluation metrics on the GOT-10k [12] dataset. The average overlap measures the average degree of the overlap between the tracking results and the ground truth annotations, reflecting the accuracy of the tracker's predictions regarding the target's locations. The success rate assesses the success detection rate of the tracker at specified thresholds, where the thresholds are set at 0.5 and 0.75. $SR_{0.5}$ and $SR_{0.75}$ represent the success rate with overlaps greater than 0.5 and 0.75, respectively. A higher SR value indicates that the tracker successfully detects the target within a larger overlapping range.

In Table 1, Precision is a measure of accuracy, and it is calculated as the Equation 1.

$$Precision = \frac{1}{f}\sum_{i=1}^{f} p(i). \tag{1}$$

Precision is calculated by taking the reciprocal (1 divided by) of the average center location error across all frames. Each frame's center location error represents how far off the predicted bounding box's center is from the ground truth bounding box's center. This error is found by calculating the Euclidean distance between these two centers for each frame. The Precision is obtained by adding up these errors for all frames and then dividing by the total number of frames (denoted as 'f').

Success measures how well the predicted bounding box overlaps with the ground truth bounding box. To calculate the Success, the reciprocal (1 divided by) of the average overlap degree is taken across all frames. The overlap degree for each frame is determined by dividing the area of intersection between the predicted bounding box and the ground truth bounding box by the area of their union. The Success metric is calculated by adding up these overlap degrees for all frames and then dividing by the total number of frames ('f').

In the dataset VOT2018 [11], visual attributes (e.g., partial occlusion, illumination changes) are annotated for each sequence, to evaluate the performance of trackers under different conditions. An evaluation system should detect errors (failures), when a tracker loses the track, and re-initialize the tracker after 5 frames following the failure for effectively utilizing the dataset. Five frames for the re-initialization are chosen, because the immediate initialization after failure leads to subsequent tracking failures. Additionally, since occlusions in videos typically do not exceed 5 frames, this setting is established. It is a distinctive mechanism to enable "reset" or "re-initialize", where a portion of frames after the reset cannot be used for evaluation.

In Table 2, the Accuracy metric evaluates how well the predicted bounding box (referred to as A_t^T) aligns with the ground truth bounding box (referred to as A_t^G) for a given frame in a tracking sequence, denoted as the t^{th} frame. This accuracy metric is symbolically represented as ϕ_t. Furthermore, $\phi_t(i,k)$ represents the accuracy of the t^{th} frame within the k^{th} repetition of a particular tracking method, where the total number of repetitions is indicated as N_{rep}. To calculate the average accuracy for this specific tracking method (i^{th} tracker), the mean accuracy over all valid frames (N_{valid}), $\rho_A(i)$, needs to be determined: $\rho_A(i)$ is computed as the sum of all $\phi_t(i)$ values divided by the total number of valid frames, N_{valid}, where t ranges from 1 to N_{valid}. The Robustness, conversely, gauges how stable a tracking method is when following a target, and a higher robustness value indicates a lower level of stability. The Robustness is quantified by using the following

Table 1. Attack performance on the dataset OTB2015

Tracker	Success				Precision			
	Original	Attack_CSA	Attack_IoU	Attack_RTAA	Original	Attack_CSA	Attack_IoU	Attack_RTAA
MixFormer	0.696	0.640	0.555	0.047	0.908	0.839	0.741	0.050
TransT	0.690	0.661	0.625	0.018	0.888	0.859	0.847	0.038

mathematical expression: $\rho_R(i)$ is calculated as the sum of tracking failures $F(i,k)$ in the k^{th} repetition of the i^{th} tracking method, divided by the total number of repetitions, N_{rep}. In Table 3, the "Failures" index counts the instances of tracking failures that occur during the tracking process of a tracking algorithm. These failures are typically related to tracking errors and do not include specific restarts or skipped frame numbers.

In Table 3, the expected average overlap (EAO), it is denoted as ϕ_{N_s}. This metric is designed to quantify the expected average coverage rate, specifically for tracking sequences up to an intended maximum length (N_s). To compute the EAO, the average intersection over union (IoU) value is considered, denoted as ϕ_i, for frames ranging from the first frame to the N_s^{th} frame in the sequence, even including the frames where tracking may have failed, and N_s represents the total sequence length. In the context of the VOT2018 [11] dataset, the calculation of expected average overlap involves taking the average of EAO values within an interval $[N_{low}, N_{high}]$, which corresponds to typical short-term sequence lengths, and the expected average overlap is denoted as $\hat{\phi}$ and is calculated by Equation 2.

$$\hat{\phi} = \frac{1}{N_{high} - N_{low}} \sum_{N_s = N_{low}:N_{high}} \hat{\phi}_{N_s}, \tag{2}$$

where the N_s ranges from N_{low} to N_{high}, and the $\hat{\phi}$ captures the expected average overlap across a range of sequence lengths, providing valuable insights into tracking performance.

In Table 4, a metric called average overlap (AO) is utilized to gauge the extent of overlap occurring during the tracking process. The AO is determined by assessing the degree of overlap for each individual frame and subsequently computing the average of these individual overlaps. The AO is the average level of overlap, and it takes the sum of the overlap values for each frame, and then it is divided by the total number of frames (N) in the sequence. Each "$Overlap_i$" represents the extent of overlap for the i^{th} frame. Additionally, Table 4 and Table 5 employ a metric known as success rate (SR) to assess how well the tracker performs under various overlap threshold conditions, and the SR quantifies the ratio of frames in which the tracker successfully keeps track of the target, considering a specific overlap threshold. The SR is a measure of how effectively the tracker follows the target. To compute it, an indicator function (I) applied to each frame's overlap value is summed up. If the overlap ($Overlap_i$) is greater than or equal to the specified threshold (Threshold), I equals 1; otherwise, it equals 0. The resulting sum is then divided by the total number of frames (N) in the sequence. For example, $SR_{0.5}$ refers to the scenario where the overlap threshold is set to 0.5, and $SR_{0.75}$ corresponds to a threshold of 0.75. These metrics offer valuable insights into how well the tracking system performs at different levels of overlap.

Experimental results are shown as follows: **Results on the dataset OTB2015 (shown in Table 1 and Figure 2).**

The original results shown in Table 1 and Figure 2, along with the results under three types of adversarial attacks, are compared. It is observed that all three attacks have certain impacts. In terms of success rate and precision, the white-box attack RTAA performed the best, causing the decrease of 93.2% and 97.4% in success rate and the drop of 94.5% and 95.7% in precision for MixFormer and TransT, respectively. The next is the black-box attack IoU, which resulted in the success rate decrease of 20.3% and 9.4%, and the precision decrease of 18.4% and 4.6% for MixFormer and TransT, respectively. Finally, the impact of the semi-black-box attack CSA, trained by SiamRPN++, is the least pronounced, with

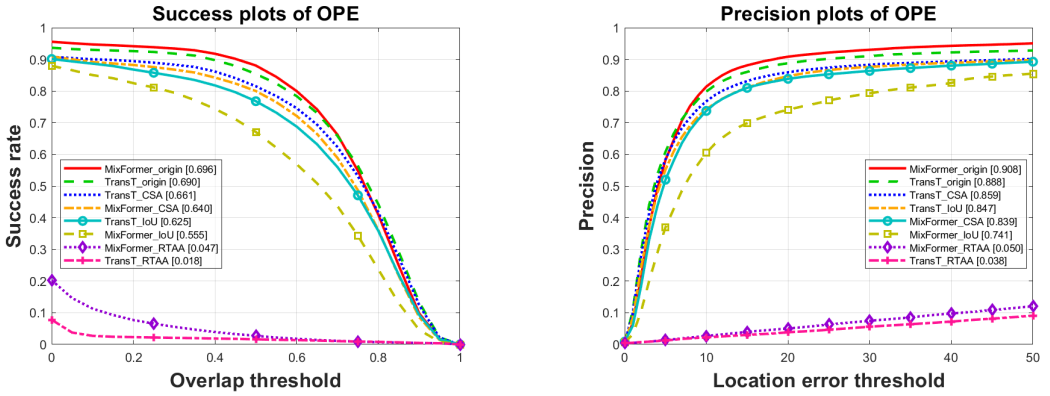

Figure 2. Evaluation results of trackers with and without adversarial attacks on the dataset OTB2015.

Table 2. Attack performance on the dataset VOT2018 (Accuracy and Robustness)

Tracker	Accuracy				Robustness			
	Original	Attack_CSA	Attack_IoU	Attack_RTAA	Original	Attack_CSA	Attack_IoU	Attack_RTAA
MixFormer	0.614	0.625	0.599	0.198	0.698	0.819	1.288	10.339
TransT	0.595	0.592	0.578	0.111	0.337	0.323	0.899	5.984

minimal influence on the tracking results. When attacking the MixFormer and TransT models, they are based on the transformer framework, and their success rates are dropped by 8.0% and 4.2%, and their precision values are decreased by 7.6% and 3.2%, respectively.

Results on the dataset VOT2018 (shown in Table 2, Table 3 and Figure 3).

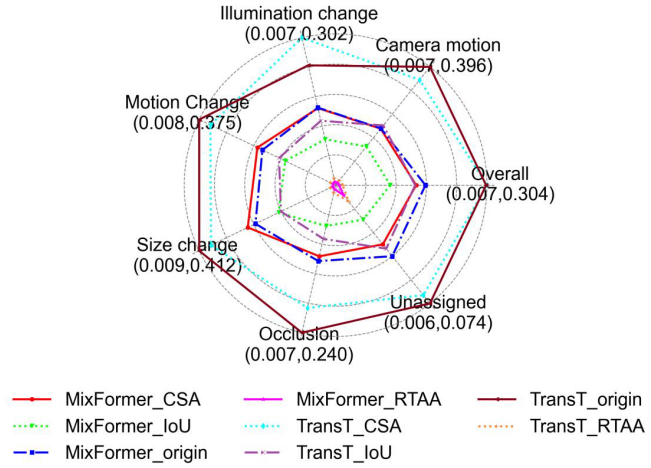

Figure 3. Quantitative analysis of different attributes on the dataset VOT2018.

As shown in Table 2, the RTAA attack achieves the best performance, followed by the IoU attack, and the CSA attack has the lowest effectiveness. Specifically, both trackers' accuracies are significantly reduced after being subjected to adversarial attacks, indicating a noticeable deviation between the tracking results after adversarial attacks and the original results. In Table 3, ranked in the order of RTAA, IoU, and CSA adversarial attacks, the main

Table 3. Attack performance on the dataset VOT2018 (Failures and EAO)

Tracker	Failures				EAO			
	Original	Attack_CSA	Attack_IoU	Attack_RTAA	Original	Attack_CSA	Attack_IoU	Attack_RTAA
MixFormer	149	175	275	2208	0.180	0.162	0.110	0.007
TransT	72	69	192	1278	0.302	0.304	0.160	0.014

Table 4. Attack performance on the dataset GOT10k (AO (%) and $SR_{0.5}$(%))

Tracker	AO(%)				$SR_{0.5}$(%)			
	Original	Attack_CSA	Attack_IoU	Attack_RTAA	Original	Attack_CSA	Attack_IoU	Attack_RTAA
MixFormer	0.716	0.680	0.554	0.048	0.815	0.768	0.629	0.037
TransT	0.720	0.702	0.529	0.046	0.821	0.798	0.609	0.051

metric EAO scores for MixFormer decrease by 96.1%, 38.9%, and 10%, respectively, while for TransT, they decrease by 95.4%, 47%, and 0%.

Figure 3 presents the performance of different attributes on the VOT2018 [11] dataset, comparing the tracking results under three types of adversarial attacks with the original results in various specific scenarios. In the radar chart, the closer a point is to the center, the worse the algorithm performs on the attribute, while points farther from the center indicate better performance.

Upon observing the target radar chart on the VOT2018 [11] dataset, a decline is clear in tracking performance when facing the three types of adversarial attacks, including scenarios involving occlusion, unassigned and overall. Among them, the RTAA attack has the strongest effect, as it exhibits nearly the worst performance in all scenarios, where the preselected box does not cover the tracking target. The IoU attack comes next, showing a comprehensive performance decrease across all scenarios. As for the CSA attack, it exhibits enhancement in certain scenarios, because the CSA attack mainly targets the SiamRPN++ model and exhibits significant attack effectiveness on this model. It means that the transferability of the CSA attack is not good to TransT and MixFormer models.

Results on the dataset GOT10k (shown in Table 4, Table 5 and Figure 4).

Table 5. Attack performance on the dataset GOT10k ($SR_{0.75}$(%))

Tracker	$SR_{0.75}$(%)			
	Original	Attack_CSA	Attack_IoU	Attack_RTAA
MixFormer	0.687	0.633	0.428	0.013
TransT	0.680	0.661	0.433	0.021

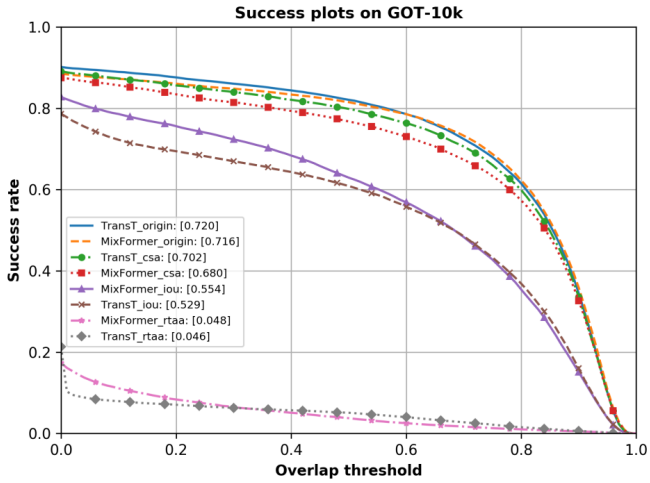

Figure 4. Evaluation results of trackers with or without adversarial attacks on the dataset GOT10k.

As shown in Table 4, Table 5 and Figure 4, three types of adversarial attacks on both trackers are conducted on the GOT-10k [12] dataset. By observing the metrics of average overlap (AO), success rate at 0.5 overlap ($SR_{0.5}$), and success rate at 0.75 overlap ($SR_{0.75}$), it is evident that the overall performance of these trackers has been decreased. Specifically, the MixFormer and TransT trackers experience a decline in the average overlap (AO) of 93.3%, 22.6%, 5.0%, and 93.6%, 26.5%, and 2.5% under the RTAA attack, the IoU attack, and the CSA attack, respectively.

Acknowledgments: This research was funded by the Key Research and Development Program of Zhejiang Province No. 2023C01141, and the Science and Technology Innovation Community Project of Yangtze River Delta No. 23002410100.

References

1. Jia, S.; Ma, C.; Song, Y.; Yang, X. Robust tracking against adversarial attacks. In Proceedings of the European Conference on Computer Vision. Springer, 2020, pp. 69–84.
2. Guo, Q.; Xie, X.; Juefei-Xu, F.; Ma, L.; Li, Z.; Xue, W.; Feng, W.; Liu, Y. Spark: Spatial-aware online incremental attack against visual tracking. In Proceedings of the European conference on computer vision. Springer, 2020, pp. 202–219.
3. Chen, X.; Yan, X.; Zheng, F.; Jiang, Y.; Xia, S.T.; Zhao, Y.; Ji, R. One-shot adversarial attacks on visual tracking with dual attention. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10176–10185.
4. Yan, B.; Wang, D.; Lu, H.; Yang, X. Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 990–999.
5. Liang, S.; Wei, X.; Yao, S.; Cao, X. Efficient adversarial attacks for visual object tracking. In Proceedings of the European Conference on Computer Vision. Springer, 2020, pp. 34–50.
6. Nakka, K.K.; Salzmann, M. Temporally-transferable perturbations: Efficient, one-shot adversarial attacks for online visual object trackers. *arXiv preprint arXiv:2012.15183* **2020**.
7. Zhou, Z.; Sun, Y.; Sun, Q.; Li, C.; Ren, Z. Only Once Attack: Fooling the Tracker with Adversarial Template. *IEEE Transactions on Circuits and Systems for Video Technology* **2023**.
8. Jia, S.; Song, Y.; Ma, C.; Yang, X. Iou attack: Towards temporally coherent black-box adversarial attack for visual object tracking. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6709–6718.
9. Yin, X.; Ruan, W.; Fieldsend, J. Dimba: discretely masked black-box attack in single object tracking. *Machine Learning* **2022**, pp. 1–19.
10. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 2411–2418.
11. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; ˇCehovin Zajc, L.; Vojir, T.; Bhat, G.; Lukezic, A.; Eldesokey, A.; et al. The sixth visual object tracking vot2018 challenge results. In Proceedings of the Proceedings of the European conference on computer vision (ECCV) workshops, 2018, pp. 0–0.
12. Huang, L.; Zhao, X.; Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence* **2019**, *43*, 1562–1577.
13. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer tracking. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8126–8135.
14. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. Mixformer: End-to-end tracking with iterative mixed attention. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 13608–13618.
15. Choi, J.; Kwon, J.; Lee, K.M. Deep meta learning for real-time target-aware visual tracking. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 911–920.
16. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable siamese attention networks for visual object tracking. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6728–6737.
17. Du, F.; Liu, P.; Zhao, W.; Tang, X. Correlation-guided attention for corner detection based visual tracking. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6836–6845.
18. Wang, N.; Zhou, W.; Wang, J.; Li, H. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1571–1580.
19. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning spatio-temporal transformer for visual tracking. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10448–10457.
20. Gao, S.; Zhou, C.; Ma, C.; Wang, X.; Yuan, J. Aiatrack: Attention in attention for transformer visual tracking. In Proceedings of the European Conference on Computer Vision. Springer, 2022, pp. 146–164.
21. Ye, B.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Joint feature learning and relation modeling for tracking: A one-stream framework. In Proceedings of the European Conference on Computer Vision. Springer, 2022, pp. 341–357.
22. Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; Ling, H. Swintrack: A simple and strong baseline for transformer tracking. *Advances in Neural Information Processing Systems* **2022**, *35*, 16743–16754.
23. Hu, X.; Liu, H.; Li, S.; Zhao, J.; Hui, Y. TFITrack: Transformer Feature Integration Network for Object Tracking. *International Journal of Computational Intelligence Systems* **2024**, *17*, 107.
24. Zhao, M.; Okada, K.; Inaba, M. Trtr: Visual tracking with transformer. *arXiv preprint arXiv:2105.03817* **2021**.
25. Bhojanapalli, S.; Chakrabarti, A.; Glasner, D.; Li, D.; Unterthiner, T.; Veit, A. Understanding robustness of transformers for image classification. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10231–10241.
26. Gu, J.; Tresp, V.; Qin, Y. Are vision transformers robust to patch perturbations? In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII. Springer, 2022, pp. 404–421.

27. Fu, Y.; Zhang, S.; Wu, S.; Wan, C.; Lin, Y. Patch-fool: Are vision transformers always robust against adversarial perturbations? *arXiv preprint arXiv:2203.08392* **2022**.
28. Bai, J.; Yuan, L.; Xia, S.T.; Yan, S.; Li, Z.; Liu, W. Improving vision transformers by revisiting high-frequency components. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV. Springer, 2022, pp. 1–18.
29. Wu, B.; Gu, J.; Li, Z.; Cai, D.; He, X.; Liu, W. Towards efficient adversarial training on vision transformers. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII. Springer, 2022, pp. 307–325.