WIP: Towards the Practicality of the Adversarial Attack on Object Tracking in Autonomous Driving

Chen Ma
Xi'an Jiaotong University
ershang@stu.xjtu.edu.cn

Ningfei Wang
University of California, Irvine
ningfei.wang@uci.edu

Qi Alfred Chen
University of California, Irvine
alfchen@uci.edu

Chao Shen✉
Xi'an Jiaotong University
chaoshen@xjtu.edu.cn

***Abstract*—Recently, adversarial examples against object detection have been widely studied. However, it is difficult for these attacks to have an impact on visual perception in autonomous driving because the complete visual pipeline of real-world autonomous driving systems includes not only object detection but also object tracking. In this paper, we present a novel tracker hijacking attack against the multi-target tracking algorithm employed by real-world autonomous driving systems, which controls the bounding box of object detection to spoof the multiple object tracking process. Our approach exploits the detection box generation process of the anchor-based object detection algorithm and designs new optimization methods to generate adversarial patches that can successfully perform tracker hijacking attacks, causing security risks. The evaluation results show that our approach has 85% attack success rate on two detection models employed by real-world autonomous driving systems. We discuss our potential next step for this work.**

I. Introduction

Object detection and object tracking are both important components of the visual perception pipeline of autonomous driving (AD) systems, which are often used to detect surrounding obstacles. Due to the development of deep neural networks (DNNs), DNN-based object detection has become a mainstream approach.

Since researchers discovered that DNNs are vulnerable to adversarial example attack [1], [2], [3], some research works in adversarial machine learning has focused on attacking object detection models of AD system in the physical world, such as using malicious patches [4] or camouflage [5] to make object detection models ignore obstacles or misclassify traffic signs. While these previous works have shown security threats to object detection, the fact that the complete visual perception pipeline of AD builds trackers for detected objects to estimate their velocities and build moving trajectories called Multiple Object Tracking (MOT), which provides robustness against previous object detection attacks. For a completed perception pipeline, attacks purely targeting object detection need to succeed over 60 consecutive frames [6], which is infeasible for current object detection attacks [4], [7], [8].

Symposium on Vehicles Security and Privacy (VehicleSec) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-88-6
https://dx.doi.org/10.14722/vehiclesec.2023.23063
www.ndss-symposium.org

Unlike previous object detection attacks that focus only on the confidence or classification scores of the object detection model, our attack method changes the position of the detected object to affect the subsequent object tracking process, and finally successfully hijacks the tracker to move in the specified direction, called tracker hijacking attack.

In this paper, we propose a novel attack for the complete visual perception pipeline for AD systems, including object detection and multiple object tracking. Our preliminary results show that we can achieve 85% attack success rate on the YOLO v3 [9] and a camera-based object detection model in Baidu Apollo [10]. We also have some comparison with existing attacks on object tracking which is detailed in §II-B.

In the future, we plan to perform more comprehensive evaluations including the effectiveness and generality, improve the practicality of the physical adversarial patch, and explore the defenses, which is detailed in §VI.

II. Background and Related Work

A. Camera-based Object Detection and Physical Attack

Camera-based perception in the AD system leverages DNN-based object detection algorithms to detect nearby objects (e.g., cars, pedestrians, and cyclists) including their position and class. The two mainstream object detection algorithms are the one-stage architecture, e.g, YOLO [11], [9] and the two-stage architecture, e.g., R-CNN [12], [13], which all adopt the anchor-based approach. To improve detection accuracy and efficiency, the anchor-based approach divides the camera image into multiple grids with fixed sizes which include a certain number of priori anchor boxes at different scales to detect different objects with different sizes. The detection network outputs the probability of the existence of an object and the corresponding object positions for each anchor box.

Recently, several existing works [4], [7], [8], [14], [15], [16] attacking object detectors in the physical space focus on generating more robust and practical adversarial examples. Meanwhile, there are many works that have tried more flexible or stealthy ways of attacking such as SLAP [17] and UPC attack [5]. However, due to object tracking, AD perception could be highly robust to these attacks purely against object detection [18], [19], which makes it difficult to affect the AD system. Thus, in this paper, we plan to conduct a physical adversarial attack on object detection with the consideration

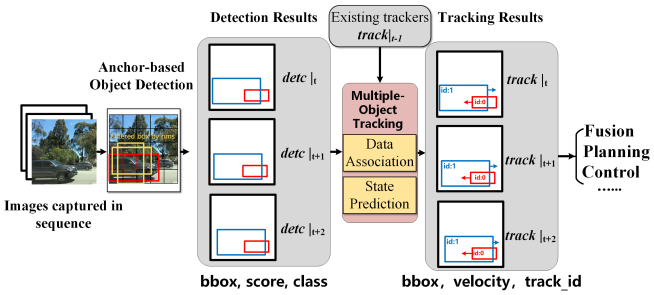


Fig. 1. An overview of an AD perception pipeline (including anchor-based object detection and tracking).

(a) Case I: moving-in attack (b) Case II: moving-out attack

Fig. 2. Illustration of our two attack goals: moving-in and moving-out attacks (in §III).

of the object tracking pipeline to achieve two unique attack goals (§III) in the physical world.

B. Multiple Object Tracking (MOT) and Attacks

MOT algorithms aim to estimate the velocity of objects in consecutive video frame sequences in order to predict the size and position of objects in subsequent frames. Due to the excellent performance of object detection, tracking-by-detection [20] has become the mainstream method of MOT, which is widely adopted in current AD systems [10], [21], and a more detailed illustration is in Fig. 1. As shown, the tracking-by-detection method associates the object detection results at time t with the existing trackers ($track|_{t-1}$) and predicts the trackers at current time t ($track|_t$) which contains velocity and position of every tracked object. To prevent detectors from false alarms and missed detections, MOT usually creates a tracker for an object only when it is continuously detected for H frames, and deletes its tracker only when the object continuously disappears for R frames [22], [10], [21], [6]. Thus, purely attacking the object detection alone may not be enough to cause damage to the entired AD system [18], [6].

Recently, several attacks on object tracking are proposed as well. For instance, Wiyatnm et al. [23] first presented an adversarial attack method for tracking tasks leveraging physical adversarial textures to cause regression-based tracking models to break away from their tracked targets. Recent works [24], [25], [26], [27] focus on the attack against Siamese-based single object trackers [28], [29], [22], and from them, the most representative one AttrackZone [27] proposed a physical attack and demonstrated it in the physical world using a projector to project adversarial perturbation into the environment. However, due to leveraging the projector, their attack usually can be only applied at night, which is less flexible, and to the best of our knowledge, none of the AD systems directly apply the Siamese-based single object tracker [10], [21], which indicates that it is unclear whether their studied targets are useful or not for AD system. Currently, state-of-the-art AD systems adopt MOT instead of the single object tracker [10], [21], [30], [31] since multiple objects must be identified and tracked concurrently in AD systems [30]. Thus, in this paper, we leverage the physical adversarial patch to attack the MOT in the AD system to overcome the two limitations above.

III. Attack Goal and Threat Model

Attack Goal. The persistent adversarial impact of an attack may cause serious security consequences such as crashing, and we consider two attack goals that can lead to safety accidents, i.e., move-in attack (Fig. 2a) and move-out attack (Fig. 2b). The goal of the move-in attack is to make the AD vehicle make an emergency stop or change to a wrong road since the attacker can move the detection results to the lane where the AD vehicle drives. For the move-out attack, it will move the target vehicle tracker to the sides of the road to make the AD vehicle believes that there are no more objects ahead, which will cause the AD vehicle to crash into the target vehicle.

Threat model. To achieve the attack goals above, in this paper, we focus on white-box adversarial attack, which allows the attacker to have access to an anchor-based object detector in the target AD system (including its structure and parameters) for generating adversarial attacks. This is the same assumption made in most prior adversarial attacks on AD perception [8], [14], [32], [33], [34]. To successfully hijack the tracker, we use dynamic adversarial patches, which means that the attacker could use different adversarial patches at different frames. This is feasible in the physical world, e.g., by displaying the patch using a monitor placed on the back side of the object [35] or projecting the patch into the scene using a projector [17].

IV. Tracker Hijacking Attack Generation

To generate adversarial patches that can achieve our attack goal in §III, i.e., change the position of the detection bounding boxes, previous work [6] simply used the difference in shape and position between the predicted and expected boxes as a loss function and minimized the loss function using an optimizer, but they did not evaluate the effectiveness of their adversarial patch generation method. Since the anchor-based object detector generates many bounding boxes for each grid of the image and filters the bounding boxes using Non-Maximal Suppression (NMS), we investigate these processes of calculating bounding boxes and propose a novel adversarial patch generation method.

In order to make the adversarial bounding box satisfy the attack success condition in §III, the position and shape of the bounding box of the target object both need to be changed. Taking YOLO and its variants as an example, the detector will apply an offset to the upper left corner of the mesh grid

to get the position of the possible objects of this grid, and slightly adjust the size of priori anchor boxes to get the size. Therefore, the position of the bounding box mainly depends on the position of the grid and the size depends mainly on the size of the priori anchor boxes. In short, to fabricate a bounding box that meets the success condition of the attack, the generation method should be able to generate an adversarial patch by picking the right anchor box in the right grid and making it be kept after NMS.

As detailed in Algorithm 1, given a target bounding box position b_t, the algorithm first finds the anchor box that is closest to its size in the grid to which the target box belongs. Then, it needs to increase the confidence score of the selected anchor box so that it can be kept after NMS, and fine-tune its size and position so that it fits the target box position as closely as possible. Similar to the existing adversarial attacks against object detection models [4], [7], [8], we also formulate the adversarial patch generation as an optimization problem. The optimization of this attack is a multiple-objective problem because it needs to optimize not only the score loss of the selected box but also the size and position loss. Given an input image x, target bounding box b_t, selected anchor box b_s, object detector without non-maximal suppression $D(\cdot)$, the optimization problem is describe as follows:

$$\arg\min_{\Delta} L_r(x+\Delta, b_t, b_s, D) \text{ such that } b_s \in B' \quad (1)$$

where Δ is the adversarial patch, L_r is the bounding box regression loss and we adopt the natural logarithm of IOU as the regression loss, B' is all detection bounding box after NMS.

To satisfy the condition in equation 1, the confidence scores of all bounding boxes whose IOU between the selected anchor box is greater than the NMS threshold must be lower than the confidence score of the selected anchor box. So the score of the selected box needs to be increased and the score of the other nearby boxes needs to be decreased. Therefore, we define the score loss as follows:

$$L_s = \lambda \cdot L_c(x+\Delta, b_s, D) - \sum_{i=0}^{B} \mathbb{1}_i^{obj} \cdot L_c(x+\Delta, b_i, D) \quad (2)$$

where λ is a fixed weight for the binary cross entropy loss L_c, $\mathbb{1}_i^{obj}$ identifies all bounding boxes B before NMS, whose IOU with the selected box is greater than the NMS threshold, if greater then $\mathbb{1}_i^{obj} \leftarrow 1$, otherwise $\mathbb{1}_i^{obj} \leftarrow 0$.

To solve the optimization problem, previous adversarial attack methods use the standard Lagrangian relaxation method, to sum up different loss functions multiplied by fixed weights and use gradient descent to find the optimal solution, which requires tuning weights to work well. This approach doesn't work well for our optimization problem, because the gradient of two losses is not balanced and the presence of mutations in the score loss optimization makes the regression loss over-optimized. And the fractional loss is not as low as the better, it only needs to satisfy the condition in equation 1, i.e all boxes that satisfy $\mathbb{1}_i^{obj} = 1$ have a lower score than the score of the selected box.

Instead of using the standard Lagrangian relaxation method to convert the task of minimizing multiple losses to minimizing the weighted sum of these losses, our approach avoids the over-optimization of regression loss and score loss being ignored because the score loss will singly be optimized until the condition in equation 1 is satisfied. That is, we reformulate the optimization problem as follows:

$$\begin{aligned}\arg\min_{\Delta} \mathbb{1}[b_s \in B'] \cdot L_r(x+\Delta, b_t, b_s, D) \\ + \mathbb{1}[b_s \notin B'] \cdot L_s(x+\Delta, b_s, D)\end{aligned} \quad (3)$$

This method is that selectively optimizes L_s or L_r depending on if $b_s \in B'$ instead of minimizing the combination of the two loss functions, ensuring that the box being optimized is always kept after NMS.

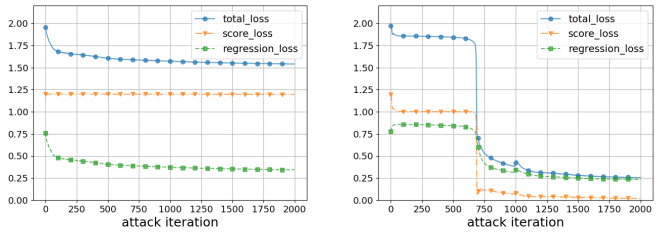
Algorithm 1 Generating Adversarial Patch

Input: Input image x, target bbox position b_t, object detector $D(\cdot)$, attack iterations N, non-maximal suppression algorithm NMS$(\cdot)$;
Output: Adversarial patch Δ.

initial Δ randomly
$(c_x, c_y) \leftarrow$ center point of b_t;
compute grid box id $grid\_id$ = compute_id(c_x, c_y);
$B = D(x+\Delta)$;
compute selected anchor box b_s = maxIoU(b_t,B(grid_id));
for $n = 0$ to N **do**
  if $b_s \in \mathrm{NMS}(D(x+\Delta))$ **then**
    $L = L_r(x+\Delta, t, b_s, D)$
  else
    $L = L_s(x+\Delta, b_s, D)$
  end if
  $\Delta = \mathrm{Adam}(\Delta, L)$
end for
return Δ

V. Preliminary Evaluation

We evaluate our attack method on two different anchor-based detectors, YOLO v3 [9] (adopted in Autoware.AI, an industry-grade full-stack AD system) and camera-based object detection model in Baidu Apollo [10] (we call it Apollo model later), both of which are representative of models for visual perception in AD systems. Moreover, we add an adversarial patch to any two selected image frames of each video frame sequence, and these two patches with different attack effects are referred to as dynamic patches in our experiments. Also, to ensure that the multiple object tracking algorithm has completed the tracker modeling process for the object, we avoid selecting the first three frames of the video frames. Our target MOT algorithm uses the following combination: the first-order Kalman filter is used to predict the position and velocity of the object the next time, and the data association between the bounding boxes takes the widely used Intersection



(a) Lagrangian relaxation method used by previous works

(b) our optimization method

Fig. 3. Comparison of losses during adversarial patch generation using different optimization method

over Union (IoU) as the similarity metric and used Hungarian matching algorithm to solve the matching problem between the currently detected bounding box and the existing tracker. Such a combination of algorithms has been adopted in previous work [6], [36] and in real-world AD systems such as Baidu Apollo [10] and Autoware.AI [21], industry-grade full-stack AD systems. We conduct the experiment on the existing dataset and our photographed data by applying the adversarial patch to the back of the target vehicle using a perspective transformation. The camera used to shoot video is the built-in camera of iPhone 12, which is set to shoot with 1080p at 30 fps. We use Adam optimizer [37] to generate the adversarial patch.

Dataset and Detection models. We selected 10 video clips from the Berkeley Deep Driving Dataset [38] with the criterion that the victim vehicle approached at a certain speed relative to the target vehicle for the object move-out scenario, which could cause a security accident once the tracker hijacking attack is successfully executed. For each clip, we manually label the patch area coordinates for each frame according to the markers of the target vehicle. We take two different detection models as the target of our attack: YOLO v3 and Apollo model. And these two models use the anchor-based approach, where YOLO v3 divides the image into three different size meshes with 3 different size anchor boxes for each grid, and the Apollo model divides the image into one mesh with 16 different size anchor boxes for each grid. For NMS, we set the score threshold at 0.4 and the threshold of IOU at 0.5.

Evaluation Metrics. In line with previous work [6], We define the success of an attack as when the detection attack ends and the detection bounding box of the target object can no longer be associated with any existing tracker. The effectiveness of the attack depends on the offset distance of the hijacked tracker before the target object is tracked again. Therefore, We use dynamic adversarial patches for a two-frame detection attack to change the target object detection bounding box, which makes the disassociated tracker move at a speed enough to cause a safety accident.

Preliminary Results. As shown in Table I, our attack method achieves 90% success rate on YOLO v3 and 80% success rate on the Apollo model. YOLO v3 is easier to attack because we can find the suitable anchor box in different sizes

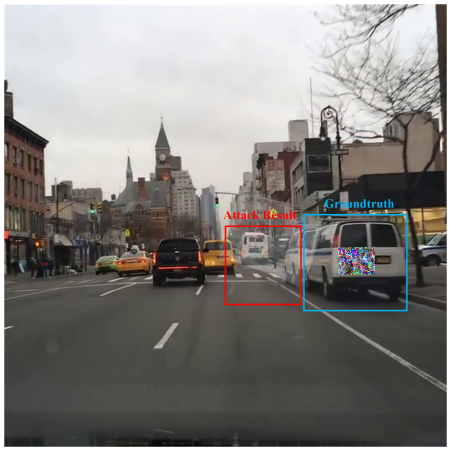


Fig. 4. Move-in attack result using dynamic adversarial patch

TABLE I
ATTACK SUCCESS RATE FOR DIFFERENT MODELS.

Target Model	YOLO v3	Apollo Model
Attack Success Rate	90%	80%

of feature maps, but the Apollo model uses only one size of the feature map, which makes it difficult to attack by our method. We also have some preliminary results and analysis compared with the methodology in the prior work [6]. As shown in Fig. 3a, the Lagrangian relaxation method used in prior works [6], [7], [4], [8] does not work well in our adversarial patch generation problem for tracker hijacking for YOLO v3. This is because the score loss exists as a sudden change phenomenon shown in Fig. 3b, which is that the gradient of score loss is small at the beginning, but decreases rapidly over dozens of iterations in the later optimization. And because there is a constraint between regression loss and score loss, it is more difficult to reduce the score loss when the regression loss is large, which makes the score loss hardly reduce in many cases. Therefore, the Lagrangian relaxation method will first optimize the regression loss for larger gradients and finally fail to satisfy the conditions in equation 1. Our optimization method solves these problems very well and achieves the results shown in Fig. 3b.

VI. Conclusion and Future Work

In this paper, we propose an adversarial attack against the complete visual pipeline of real-world AD systems which hijacks the tracker in multi-object tracking by controlling the object detection results. In particular, we propose an effective adversarial patch generation method that exploits the generation process of detection results for anchor-based object detection to control the bounding boxes of objects. Experimental results show that our method yields an average attack success rate of 85%. In addition, the evaluation results demonstrate the effectiveness of our optimization method.

In future work, we plan to perform more comprehensive evaluations including the effectiveness, generality, and comparison with baseline, improve the practicality of the physical

adversarial patch, and explore the defenses. 1) Comprehensive evaluation: we will evaluate our attack in a large-scale dataset (currently, we only evaluate in 10 video frames) to show our attack's effectiveness. In addition, we also plan to evaluate the generality of our attack methodology such as evaluate on other object types, other MOT, and MOT in different settings. Also, we will compare our work to the state-of-the-art practical tracking attack such as AttrackZone [27] from both the methodology side and the evaluation side to show the value of our work. 2) Practicality: we plan to improve the practicality and robustness of the adversarial patch to make our adversarial patch work successfully in the physical world and can have effects on the end-to-end AD systems, which is our final goal of this work. 3) Defenses: we hope that our attack can further guide future MOT designs to improve their robustness. Thus, in the future, defenses will be also a mainstream of our work.

VII. Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback. This research was supported in part by the NSF under grants CNS-2145493, CNS-1929771, CNS-1932464, USDOT under Grant 69A3552047138 for the CARMEN UTC, the ARO under contract W911NF1810208, National Key R&D Program (2020YFB1406900), National Natural Science Foundation of China (U21B2018, 62161160337, 61822309, U20B2049, 61773310, U1736205, 61802166), and Shaanxi Province Key Industry Innovation Program (2021ZDLGY01-02). Chao Shen is the corresponding author.

References

[1] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE S&P*. Ieee, 2017, pp. 39–57.

[2] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Asia CCS*, 2017, pp. 506–519.

[3] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, and T. Wang, "Interpretable deep learning under fire," in *29th {USENIX} Security*, 2020.

[4] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, "Seeing isn't believing: Towards more robust adversarial attack against real world object detectors," in *ACM CCS*, 2019, pp. 1989–2004.

[5] L. Huang, C. Gao, Y. Zhou, C. Xie, A. L. Yuille, C. Zou, and N. Liu, "Universal physical camouflage attacks on object detectors," in *CVPR*, 2020, pp. 720–729.

[6] Y. J. Jia, Y. Lu, J. Shen, Q. A. Chen, H. Chen, Z. Zhong, and T. W. Wei, "Fooling detection alone is not enough: Adversarial attack against multiple object tracking," in *ICLR*, 2020.

[7] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector," in *ECML PKDD*. Springer, 2019, pp. 52–68.

[8] W. Jia, Z. Lu, H. Zhang, Z. Liu, J. Wang, and G. Qu, "Fooling the eyes of autonomous vehicles: Robust physical adversarial examples against traffic sign recognition systems," *NDSS*, 2022.

[9] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[10] "Baidu Apollo," http://apollo.auto, 2022.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *CVPR*, 2017, pp. 7263–7271.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in neural information processing systems*, vol. 28, 2015.

[14] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, "Invisible for both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical World Attacks," in *IEEE S&P 2021*, May 2021.

[15] C. DiPalma, N. Wang, T. Sato, and Q. A. Chen, "Security of camera-based perception for autonomous driving under adversarial attack," in *IEEE S&P Workshops (SPW)*. IEEE, 2021, pp. 243–243.

[16] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, "3d adversarial object against msf-based perception in autonomous driving," in *MLSys*, 2020.

[17] G. Lovisotto, H. Turner, I. Sluganovic, M. Strohmeier, and I. Martinovic, "{SLAP}: Improving physical adversarial examples with {Short-Lived} adversarial perturbations," in *30th USENIX Security*, 2021.

[18] J. Shen, N. Wang, Z. Wan, Y. Luo, T. Sato, Z. Hu, X. Zhang, S. Guo, Z. Zhong, K. Li *et al.*, "Sok: On the semantic ai security in autonomous driving," *arXiv preprint arXiv:2203.05314*, 2022.

[19] N. Wang, Y. Luo, T. Sato, K. Xu, and Q. A. Chen, "Poster: On the system-level effectiveness of physical object-hiding adversarial attack in autonomous driving," in *ACM CCS*, 2022, pp. 3479–3481.

[20] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, p. 103448, 2021.

[21] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *ICCPS*. IEEE, 2018, pp. 287–296.

[22] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *ECCV*, 2018.

[23] R. R. Wiyatno and A. Xu, "Physical adversarial textures that fool visual object tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4822–4831.

[24] Q. Guo, X. Xie, F. Juefei-Xu, L. Ma, Z. Li, W. Xue, W. Feng, and Y. Liu, "Spark: Spatial-aware online incremental attack against visual tracking," in *ECCV*. Springer, 2020, pp. 202–219.

[25] X. Yan, X. Chen, Y. Jiang, S.-T. Xia, Y. Zhao, and F. Zheng, "Hijacking tracker: A powerful adversarial attack on visual tracking," in *ICASSP*. IEEE, 2020, pp. 2897–2901.

[26] L. Ding, Y. Wang, K. Yuan, M. Jiang, P. Wang, H. Huang, and Z. J. Wang, "Towards universal physical attacks on single object tracking." in *AAAI*, 2021, pp. 1236–1245.

[27] R. Muller, Y. Man, Z. B. Celik, M. Li, and R. Gerdes, "Physical hijacking attacks against object trackers," in *ACM CCS*, 2022.

[28] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *CVPR*, 2018.

[29] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[30] F. Leon and M. Gavrilescu, "A review of tracking and trajectory prediction methods for autonomous driving," *Mathematics*, vol. 9, no. 6, 2021. [Online]. Available: https://www.mdpi.com/2227-7390/9/6/660

[31] J. Chen, F. Wang, C. Li, Y. Zhang, Y. Ai, and W. Zhang, "Online multiple object tracking using a novel discriminative module for autonomous driving," *Electronics*, vol. 10, no. 20, p. 2479, 2021.

[32] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, "Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack," in *USENIX Security*, 2021.

[33] T. Sato, J. Shen, N. Wang, Y. J. Jia, X. Lin, and Q. A. Chen, "Hold tight and never let go: Security of deep learning based automated lane centering under physical-world attack," *arXiv:2009.06701*, 2020.

[34] ——, "Wip: Deployability improvement, stealthiness user study, and safety impact assessment on real vehicle for dirty road patch attack," in *AutoSec*, vol. 2021, 2021, p. 25.

[35] Y. Man, R. Muller, M. Li, Z. B. Celik, and R. Gerdes, "That Person Moves Like A Car: Misclassification Attack Detection for Autonomous Systems Using Spatiotemporal Consistency," in *USENIX Security*, 2023.

[36] W. Feng, Z. Hu, W. Wu, J. Yan, and W. Ouyang, "Multi-object tracking with multiple cues and switcher-aware classification," *arXiv preprint arXiv:1901.06129*, 2019.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[38] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, vol. 2, no. 5, p. 6, 2018.