FlyTrap: Physical Distance-Pulling Attack Towards Camera-based Autonomous Target Tracking Systems

Shaoyuan Xie Mohamad Habib Fakih Junchi Lu Fayzah Alshammari Ningfei Wang
Takami Sato Halima Bouzidi Mohammad Abdullah Al Faruque Qi Alfred Chen
University of California, Irvine

***Abstract*—Autonomous Target Tracking (ATT) systems, especially ATT drones, are widely used in applications such as surveillance, border control, and law enforcement, while also being misused in stalking and destructive actions. Thus, the security of ATT is highly critical for real-world applications. Under the scope, we present a new type of attack: *distance-pulling attacks* (DPA) and a systematic study of it, which exploits vulnerabilities in ATT systems to dangerously reduce tracking distances, leading to drone capturing, increased susceptibility to sensor attacks, or even physical collisions. To achieve these goals, we present *FlyTrap*, a novel physical-world attack framework that employs an adversarial umbrella as a deployable and domain-specific attack vector. FlyTrap is specifically designed to meet key desired objectives in attacking ATT drones: physical deployability, closed-loop effectiveness, and spatial-temporal consistency. Through novel progressive distance-pulling strategy and controllable spatial-temporal consistency designs, FlyTrap manipulates ATT drones in real-world setups to achieve significant system-level impacts. Our evaluations include new datasets, metrics, and closed-loop experiments on real-world white-box and even commercial ATT drones, including DJI and HoverAir. Results demonstrate FlyTrap's ability to reduce tracking distances within the range to be captured, sensor attacked, or even directly crashed, highlighting urgent security risks and practical implications for the safe deployment of ATT systems. Video demonstrations and code can be found at https://sites.google.com/view/av-ioat-sec/flytrap.**

I. Introduction

Autonomous Target Tracking (ATT), also known as *Active Track* [6] or *Dynamic Track* [3], allows autonomous systems to follow a designated target (e.g., a person) while maintaining a consistent distance [14], [15], [78], [79]. Drones [22], [21], [97], [90] have become the leading platform for ATT due to their versatility, supporting applications such as security surveillance [101], [1], border control [49], and law enforcement [52] beyond entertainment. Some of these applications are already deployed in practice, e.g., U.S. police departments are using ATT drones to track individuals for law enforcement [52]. Conversely, this technology poses significant security, privacy, and safety threats if misused in criminal scenarios, e.g., to facilitate stalking [42], or lethal/destructive actions by carrying explosives or weapons [18], [67], [89].

All these real-world applications, whether benign or criminal, make the security of ATT systems critically important. In this work, we exploit new vulnerabilities in ATT systems by causing drones to dangerously decrease their tracking distance from targets, which we define as *distance-pulling attack* (DPA). The DPA can lead to various severe physical consequences by causing the drone to be: (1) physically captured after being pulled into a reachable range (e.g., by a net gun [32], [13], [19]); (2) made much more attackable by a wider band of sensor attacks (e.g., camera spoofing [124], acoustic attacks [98], [44]), which by nature has range limitations [124], [98]; and (3) physically crashed, after the distance between the drone and the tracking target is shortened close enough to be within physically hitting distance. In contrast to other attacks on object tracking that can be possibly applied to ATT, such as those that can cause the model to lose track of the target [69], [107], our proposed DPA can enable a more fundamental elimination of the drone since the attacker can physically capture it, reverse-engineer it [95], and/or identify the underlying pilot as law enforcement evidence collection [50]. Thus, understanding the security challenges and practical implications of DPA against ATT systems, especially those that are already commercially available, is imperative.

Most modern ATT systems rely on cameras, given the cost-efficiency and ease of deployment on drone platforms [22], [21], [97], [90]. Specifically, Deep Neural Network (DNN) based Single Object Tracking (SOT) [5], [57], [17], [56] is a core step in the latest camera-based ATT systems to achieve stable target tracking as shown in Fig. 1. Prior works have demonstrated vulnerabilities in SOT models through pixel-level perturbations [31], [115], [12] or physical attacks [20], [107], [69]. However, these studies primarily focus on manipulating tracking (e.g., move-in and move-out attacks [69]) while the ATT system is composed of both tracking and distance control. Therefore, these prior works do not address the core challenges we identify for DPA against ATT systems below.

Specifically, first, practical entry points for real-world attacks on ATT systems remain under-explored. Previous tracking attacks that focused on digital perturbations [31], [115], [12], [117] often lack physical feasibility. Additionally, physical attack vectors such as TV screens [107], printed letter-size

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.230904
www.ndss-symposium.org

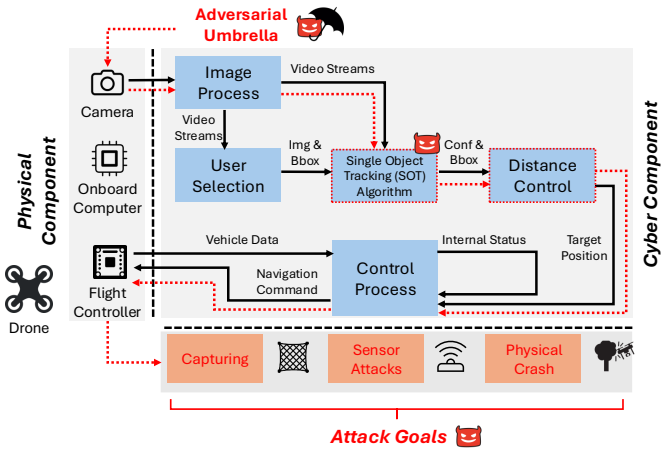


Fig. 1: Overview of the Autonomous Target Tracking (ATT) system data flow and our proposed distance-pulling attack (DPA) propagation path. We treat the camera as a physical entry point and use the adversarial umbrella to attack the Single Object Tracking (SOT) model and then the distance control algorithm to cause system-level distance-pulling effects, achieving attack goals including drone capturing, range-limited sensor attacks, or direct crashing.

papers [20], or projectors [69] face significant challenges due to their limited deployability in uncontrolled outdoor environments. Second, prior works fail to experimentally demonstrate the generalizability of their attacks across scenarios, where a single attack pattern is effective against unseen targets and/or backgrounds. Third, the newly proposed DPA against the ATT systems inherently demands closed-loop effectiveness, where current attack results influence future frames. The systems targeted by prior works [107], [115], [20], [69] do not involve distance control and thus do not consider addressing this newly raised challenge in ATT systems. Lastly, these works [115], [69] ignore the spatial-temporal consistency and can be defended by existing consistency checking-based defense methods [71].

To address these critical research challenges, we present the first systematic study on the security of camera-based ATT under a newly defined physical-world DPA. Our approach centers on three key design objectives to ensure the success and stealthiness of the attack: (i) *physical and real-world deployability*, where the attack vector physically misguides the ATT system's distance control mechanism while remaining easy to deploy, robust to lighting conditions [69], and remains inconspicuous; (ii) *closed-loop effectiveness*, where the attacks progressively shorten the tracking distance in closed-loop fashion, achieving system-level physical impacts; and (iii) *spatial-temporal consistency*, which allows the DPA to be consistent spatially and temporally, evading latest anomaly detection-based defenses [71], [66], [36], [120].

To achieve the above objectives, we introduce *FlyTrap*, a novel physical DPA against ATT systems. FlyTrap is the first to systematically tackle these challenges by utilizing *adversarial umbrella* as a novel domain-specific attack vector, i.e., a physical attack vector that an ATT-tracked target can naturally and dynamically deploy for self-coverage. The umbrella, designed for ease of carriage and inconspicuous deployment, can be naturally oriented upward toward the drone. In the ATT system context, using such an attack vector can simultaneously offer advantages in physical realizability and real-world deployability as desired above. Additionally, we design a novel progressive distance-pulling strategy, enabling continuous distance-pulling under closed-loop control. We further design our attack to maintain spatial-temporal consistency, which can bypass current state-of-the-art consistency cross-checking-based defense mechanisms [71], [66], [36], [120]. Our approach combines novel attack vectors, progressive distance-pulling, and a controllable design for spatial-temporal consistency to achieve physical, real-world deployable, closed-loop, effective, and spatial-temporal consistent attacks.

In evaluation, we construct new datasets and introduce metrics to evaluate the system-level impact of the proposed DPA, which shows high effectiveness, scenario universality, and spatial-temporal consistency. In physical experiments, we craft real-world adversarial umbrella prototypes optimized on different white-box models. Then, we implement a full-stack ATT drone from scratch. The experimental setups provide a closed-loop evaluation to understand the physical impact of our FlyTrap design under the white-box assumption. Our white-box, closed-loop physical experiments show that FlyTrap can achieve 100% success rate in pulling the drone close enough to induce capturing, sensor attacks, and/or direct crashes. To further assess the real-world impacts, we conduct black-box DPA against three commercial drones: the DJI Mini 4 Pro, the DJI NEO, and HoverAir X1. The results show that our newly proposed DPA can indeed cause system-level DPA attack effects on them. We further show end-to-end FlyTrap-enabled DPA demonstrations against these commercial drones, leading to their capture or crash, demonstrating DPA's strong applicability in real-world attack scenarios. We also investigate the stealthiness of FlyTrap-optimized patterns by conducting a user study with 200 participants, and further discuss potential countermeasures. Video demonstrations and code can be found on our project website at **https://sites.google.com/view/av-ioat-sec/flytrap**. To summarize, our contributions include:

- **Problem formulation**: We are the first to define distance-pulling attacks (DPA) of camera-based ATT drones. We formally define the problem with domain-specific objectives and introduce the adversarial umbrella as a novel, physically deployable attack vector.
- **Novel design**: We propose FlyTrap, including a progressive distance-pulling strategy and a controllable spatial-temporal consistency design, encompassed by an end-to-end optimization pipeline for attacking ATT drones.
- **Evaluation**: We construct a new dataset and define system-level metrics for comprehensive evaluation. The results highlight our design to achieve closed-loop and spatial-temporal consistent attacks.
- **Physical-world impact**: We implement full-stack ATT

drones, craft physical adversarial umbrellas, and conduct end-to-end evaluations in real-world setups, showing direct system-level impact. We further performed extensive black-box testing on three commercial drones, showing high real-world applicability of the proposed attacks.

II. Background and Problem Formulation

A. Camera-based Autonomous Target Tracking (ATT) Drone

Fig. 1 illustrates a typical camera-based ATT system [121], [10], [51], [30], which follows a hierarchical control architecture consisting of an inner and an outer loop [10]. The inner loop, integrated into the flight controller, handles low-level flight stability and receives navigation commands. The outer loop manages high-level perception and decision-making tasks, including image processing, object tracking, distance estimation, and flight path planning.

Single Object Tracking (SOT) algorithm. The SOT algorithm is a crucial component in the ATT pipeline, primarily responsible for generating navigation commands. Contemporary SOT algorithms are predominantly based on Deep Neural Networks (DNN) [5], [57], [17], [56]. The SOT model uses a *template frame* as a reference and predicts the target's location in *search frames*, as illustrated in Fig. 2. The target tracking task can be formulated as a conditional prediction, as shown in the equation below:

$$\{(cx_j, cy_j, w_j, h_j, score_j)\}_{j=1}^{M} = F(\mathbf{I}_{\text{search}} | \mathbf{I}_{\text{tplt}}), \quad (1)$$

where F denotes the SOT model, $\mathbf{I}_{\text{search}}$ and $\mathbf{I}_{\text{tplt}}$ denote the search frame and template frame, respectively. $\mathcal{P}_j = (cx_j, cy_j, w_j, h_j)$ denotes the localization results, including the x- and y-axis center coordinates, width, and height. $score_j$ denotes the prediction confidence. M represents the number of prediction proposals. The proposal with the highest confidence is regarded as the final tracking output.

Distance Control. This component estimates the relative distance to the tracked object using the SOT output and translates it into flight control actions (e.g., next waypoint). The most widely adopted strategy in real-world systems today is 2D-based distance control [10], [51], [30], which infers navigation commands directly from the 2D object bounding box. More specifically, in Fig. 2, the drone adjusts its yaw, roll, and/or altitude to center the bounding box within the current frame and moves forward or backward to maintain the bounding box size, thereby preserving a stable tracking distance. This system design motivates our formulation of DPA as a fundamental system-level attack objective for ATT. Specifically, we strategically shrink the tracking bounding box to deceive the drone into perceiving that the object is moving away, thus moving closer for compensation, leading to reduced tracking distance.

B. Problem Formulation

While disrupting the SOT component to lose track of the target can temporarily disable the ATT functionality [115], [69], [107], it does not fundamentally prevent the system from resuming tracking, either through manual re-selection or

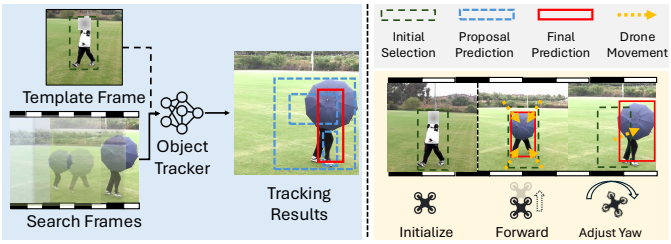


Fig. 2: *Left*: Single Object Tracking (SOT) depends on the initialization as the template frame and tracks the target in search frames. *Right*: The drone adjusts its position to keep the box at the center and the same size as the template frame.

operator intervention. As shown in the ATT system pipeline (Section II-A), the ATT system operates based on both SOT and distance control for maintaining a stable tracking distance. From this system perspective, we focus on exploiting vulnerabilities in the position control mechanism. A particularly compelling attack objective—and the focus of this work—is to intentionally reduce the tracking distance, pulling the ATT drone dangerously close to the tracked target, which we define as the distance-pulling attack (DPA). As shown in Fig. 3, DPA can be exploited to achieve *A1*: drone capturing, e.g., by using a net gun [19] (shown in Section V-G); *A2*: range-limited sensor attacks [98], [124]; or *A3*: causing the drone to crash into the target (also shown in Section V-G). In either case, this can result in a more permanent elimination of tracking capabilities, compared to losing tracking [115], [69].

Considering both the benign and criminally motivated applications of ATT (Section I), the incentives for this overall attack goal can also be either benign or malicious. For example, when used against benign applications (e.g., security surveillance [101], border control [49], and law enforcement [52]), the attack incentives are malicious and can directly threaten public security by capturing the drone and exploiting vulnerabilities for future counter-measures. However, when used for criminally-motivated scenarios (e.g., stalking [42] or lethal actions [18], [67], [89]), the attack incentives may be benign, empowering individuals to defend themselves, e.g., by capturing unauthorized drones, identifying the pilot, and extracting flight logs to uncover malicious intent [77], [76]. Thus, although we generally call it an "attack" in this paper, the security problem studied can be exploited for social good, and the "attacker" may be non-malicious individuals who just want to protect their privacy and safety.

C. Threat Model

In this paper, we mainly target ATT drone setups that perform tracking within 20 meters, which is the most typical ATT operation range for person tracking for consumer drones today (e.g., DJI Mini [25], Potensic [81], Autel [4], Skydio 2 [26]) and also is a range that can more easily allow the attacker to notice the tracking and thus launch the attack. Note that our attack is not limited to this range by design; the attack distance and angle practicality are further discussed in

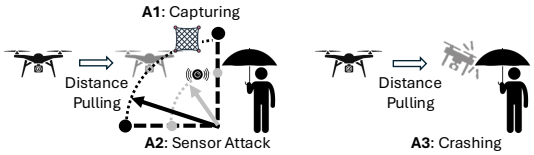


Fig. 3: Illustration of distance-pulling attack (DPA) and attack goals targeted in this work. We target to dangerously shorten the tracking distance of ATT drones, which can be exploited to cause the drone to be *A1*: captured; *A2*: under range-limited sensor attacks; or even *A3*: crashed into the attack umbrella.

Appendix B. We start with a white-box attack design setup, i.e., we assume that the attacker has full knowledge of the SOT model used in the victim ATT system. This can be accomplished by first collecting information about the targeted drones with the ATT feature [22], [21], [97], [90] and then purchasing the same model and reverse engineering it, which is feasible given recent advances in reverse-engineering such systems [95] and machine learning models [108], [62]. We also assume that the attacker can collect videos of different tracking scenarios, but these videos are not necessarily for the same tracking scenario during the attack (i.e., for the same tracking target instance and/or background location when the attack is launched), as we show in the scenario universality evaluation (Section V-C).

Although our method is developed under a white-box assumption, it can potentially be extended to black-box settings by leveraging the transferability of adversarial patterns [61]. As black-box settings are more practical, we also evaluate them by performing (1) attack transferability evaluation across open-source models (Section V-D), and (2) direct black-box testing on commercial ATT drones (Section V-G).

III. Related Works and Design Challenges

A. Related Works and Comparisons

Autonomous systems security. Security research on autonomous systems primarily falls into two categories: sensor security and AI security. For sensor security, prior work has examined threats to commonly used sensors in autonomous systems, including cameras [44], [8], [116], LiDAR [92], [80], [9], [8], gyroscopic [98], IMU [102], etc. In contrast, autonomous AI security research has primarily targeted self-driving vehicles, such as traffic sign recognition systems [123], [47], [105], [94], automatic lane centering [93], [48], high-autonomy autonomous driving systems [8], [92], [104], [122], etc. Recently, Zhou et al. were among the first to investigate autonomous AI security in drone contexts, with a focus on stereo camera-based collision avoidance [124]. To the best of our knowledge, we are the first to propose and conduct a system-level security analysis of DPA in camera-based ATT.

Adversarial attacks on SOT. While we are the first to propose DPA in ATT systems, prior work has examined vulnerabilities in SOT models individually [31], [115], [12], [117], [58], [46], [72], [107], [20], [11], [69]. Specifically, various prior works explored using pixel perturbation to attack SOT models [31], [115], [12], [117], [58], [46], [72]. However, these studies focus on offline video processing rather than real-time ATT systems, and therefore do not address: (1) physical-world deployment, (2) effective attack across closed-loop ATT control, and (3) spatial-temporal consistency.

Some more recent prior works have started to consider physical-world attack vectors [107], [20], [11], [69]. However, their attack vectors: TV screen [107], printed paper [20], [11], and projectors [69] face serious challenges for practical deployment for the following reasons: printed paper is barely visible from an aerial perspective; TV screen [107] has limitations during the carrying phase; and projectors used in AttrackZone are subject to lighting conditions and require a close enough flat surface for projection, as acknowledged by the authors [69]. Moreover, ATT systems generally operate in well-lit, outdoor environments where attackers have limited control, making projection-based attacks difficult to execute reliably. Moreover, these works were not designed with DPA and closed-loop ATT systems in mind. As a result, these approaches overlook the closed-loop dynamics critical to achieving better distance-pulling effects. Last but not least, the advanced consistency-checking defense can already detect these attacks [71], given their insufficient spatial-temporal consistency considerations.

B. Design Challenges

Based on the above analysis, we identify key challenges in designing DPA against ATT systems.

C_1**: Physical and real-world deployable attack vectors for ATT systems.** Designing effective attacks against ATT systems requires physical, highly deployable vectors. Prior attack vectors, while effective in controlled environments, face significant limitations when deployed in real-world ATT settings [107], [69], [20] as discussed in Section III-A. The attackers often have minimal control over environmental factors, especially when they are unwillingly tracked outdoors. This highlights the need for a more versatile, inconspicuous, and deployable physical attack vector.

C_2**: Closed-loop distance-pulling effects.** A successful DPA must sustain closed-loop distance-pulling effects. In the context of ATT systems, this means the attack at the current frame will influence the drone's behavior in subsequent frames, reducing the tracking distance in a feedback loop. However, existing works adopt an open-loop[1] approach, where attacks are optimized independently from the ATT system's response [107], [69], [20]. Such methods fail to address the dynamic and autonomous nature of drone tracking, where attacks must continuously pull the drone closer in response to reduced tracking distances. The motion model in DRP [93], designed for lane-centering in ground vehicles, does not generalize to the aerial dynamics of drone tracking.

C_3**: Spatial-temporal consistency.** Attacking object tracking introduces additional challenges compared to object de-

[1]The open-loop concept in this paper is a similar concept from control theory. By open-loop, we mean the attacker conducts attacks without considering the control feedback loop from the victim systems.

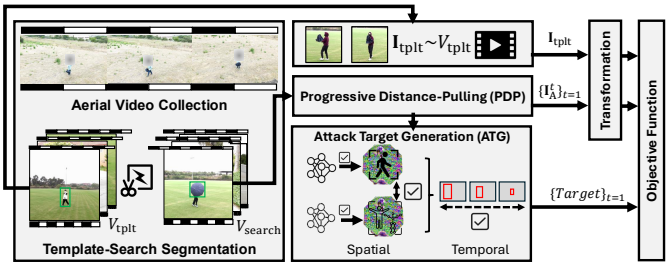


Fig. 4: FlyTrap overall pipeline. We design an adversarial umbrella as a domain-specific and deployable attack vector. The progressive distance-pulling (PDP) achieves the closed-loop distance-pulling effects while the attack target generation (ATG) constrains the spatial-temporal consistency.

tection due to the inherent spatial-temporal consistency of tracking algorithms [69]. Additionally, recent state-of-the-art consistency-based defenses have demonstrated promising performance in securing vision-based autonomous systems [66], [71], [36], [114], [120], further increasing the difficulty of maintaining consistency in attacks. Although prior work has considered the spatial-temporal-based defense [69], their methods primarily target simplistic approaches, such as Kalman filters, leaving them detectable to more advanced anomaly detection mechanisms [71]. Addressing this challenge requires developing adversarial attacks under the constraint of maintaining spatial-temporal consistency in both the tracking model and auxiliary consistency-checking mechanisms.

IV. FlyTrap

This section introduces FlyTrap, the first physical and system-level DPA targeting camera-based ATT drones. As shown in Fig. 4, to achieve the attack goal and address design challenges (Section III-B), our FlyTrap attack introduces novel designs: attack vectors, a progressive distance-pulling strategy, and controllable spatial-temporal consistency.

A. Design Overview

Adversarial umbrella: A domain-specific, physically deployable attack vector. We propose *adversarial umbrellas* as a novel class of physical attack vectors tailored for camera-based ATT drones. An umbrella is an ideal medium for adversarial patterns because: (1) it offers a large, nearly flat, rigid surface for pattern printing; (2) it naturally fits outdoor scenarios, requiring minimal setup and offering ease of transport and deployment; and (3) it offers fine control, allowing the attacker to maximize exposure and obscure themselves. Additionally, umbrellas do not require elaborate directional alignment or power sources, directly addressing challenge C_1 in the ATT drone context. In deployment, the attackers only need to cover their upper bodies and point the umbrella at the drone. While standing still is sufficient, crouching and hiding the entire body can further increase success by occluding any visible parts. Note that we don't mean to claim the physical adversarial patch as the major scientific contribution, but rather a practical delivery mechanism to support our design below.

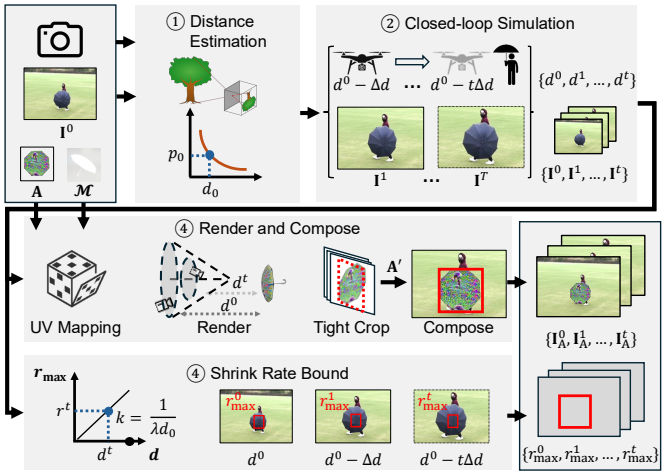


Fig. 5: Design for progressive distance-pulling via physical modeling. We propose to leverage computer graphics to simulate the closed-loop dynamics of the DPA process. We further derive the upper bound to set the shrink rate for each stage.

Progressive distance-pulling via physical modeling. To address the challenge of closed-loop effectiveness (C_2), we proposed modeling the appearance of adversarial patterns as the drone gradually approaches, simulating the effects of reducing distance under DPA. By incorporating camera geometry, physical rendering, and our proven upper-bound shrink rate setups, our design ensures that the attack remains effective as the drone approaches, ensuring consistent distance-pulling effects.

Controllable spatial-temporal consistency. To address the spatial-temporal consistency challenge (C_3), we introduce an attack target generator for adaptive attacks that jointly constrain spatial and temporal features across models and frames. The attack target generator explicitly encodes the spatial-temporal constraint by manipulating features like box shape, key points, or pose estimation within the adversarial region, simulating human-like motion and appearance. This enables us to bypass consistency-based defense systems, which are receiving growing attention in securing autonomous vehicles.

B. Progressive Distance-Pulling via Physical Modeling

In this section, we introduce our solution to challenge C_2. As shown in Fig. 5, via progressive distance-pulling (PDP), we simulate the effect through physical modeling in computer graphics, including ① distance estimation, ② closed-loop simulation, ③ rendering and composition, and our proven ④ shrink rate bound. The initial inputs are the camera model, the search frame $\mathbf{I}^0 \in \mathbb{R}^{H\times W\times 3}$, where H and W represent the height and width, the initial adversarial pattern $\mathbf{A} \in \mathbb{R}^{H_a\times W_a\times 3}$, where H_a and W_a represent the height and width of the adversarial pattern, and the umbrella 3D mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, defining vertices, edges, and faces. The output is a set of simulated images $\{\mathbf{I}^0_\mathbf{A}, \mathbf{I}^1_\mathbf{A}, ..., \mathbf{I}^t_\mathbf{A}\}$ and maximum shrink rates $\{r^0_{\max}, r^1_{\max}, ..., r^t_{\max}\}$. The shrink rate is defined

as the ratio between the attacked bounding box area and the umbrella's visible area.

In step ①, we adopt a pinhole camera model with focal length f. The relationship between the pixel length p and the actual length s is defined by: $d = \frac{f \cdot s}{p}$. The focal length can be retrieved from the camera's specifications, and with the pixel length in each image $\mathbf{I}^0$, we can estimate the distance between the drone and the object. This estimate serves as the initial distance d^0 for the subsequent closed-loop simulation.

In step ②, we simulate distance-pulling behavior as the drone incrementally approaches the target. Starting from the initial distance d^0, we iteratively reduce it using a user-defined interval: $d^t = d^0 - t\Delta d$, where t denotes the time step and Δd the distance decrement per step. Given each distance, we estimate the corresponding pixel length and synthesize a sequence of progressively zoomed-in images from $\mathbf{I}^0$, denoted as $\{\mathbf{I}^1, \ldots, \mathbf{I}^t\}$. We assume the camera is oriented directly toward the tracked object, an assumption justified by the attacker's ability to aim the umbrella at the drone.

Then, we simulate the umbrella geometry with high physical fidelity. In the rendering step, we first construct a UV mapping, which projects a 2D adversarial pattern $\mathbf{A}$ onto the 3D model. The UV mapping function $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ maps 2D coordinates $\mathbf{u}_i \in \mathbb{R}^2$ of the adversarial pattern to the corresponding 3D positions $\mathbf{v}_i \in \mathbb{R}^3$ on the mesh vertices $\mathcal{V}$. This allows the seamless attachment of the adversarial pattern onto the umbrella's surface, accounting for its curvature and topology, which is essential for maintaining real-world fidelity during optimization. We render the umbrella by placing a virtual camera at the estimated positions $\{d^0, d^1, \ldots, d^t\}$ from the mesh. The camera is oriented toward the umbrella's center, with elevation angle $\theta = 0$ and azimuth angle $\phi = 0$. We detail how camera angle randomization improves real-world robustness in Section IV-D3. After rendering, we apply image processing steps, including grayscale conversion, binarization, and morphological dilation, to perform edge segmentation and remove background margins, which produces a tightly cropped rendered image $\mathbf{A}' \in \mathbb{R}^{H'_a \times W'_a \times 3}$, facilitating seamless composition in the next stage. H'_a and W'_a represent the height and width of the rendered and cropped adversarial patterns. The overall process can be expressed as:

$$\mathbf{A}' = \text{TightCrop}(\text{Render}(\Phi(\mathbf{A}), d, \theta, \phi)). \tag{2}$$

In the composing step, we compose the rendered adversarial pattern $\mathbf{A}'$ to the target location in the simulated image $\{\mathbf{I}^0, \mathbf{I}^1, ..., \mathbf{I}^t\}$ and get the final adversarial images $\{\mathbf{I}^0_{\mathbf{A}}, \mathbf{I}^1_{\mathbf{A}}, ..., \mathbf{I}^t_{\mathbf{A}}\}$. This is achieved by computing a projection matrix followed by an affine transformation.

Finally, in step ④, we formally derive Theorem IV-B, which establishes the relationship between the shrink rate and the resulting pulling distance. Thus, given a distance d^t in the closed-loop simulation, to ensure the attack can pull the drone into the next simulated distance d^{t+1}, the maximum shrink rate at step t should be $r^t_{\max} = \frac{d^{t+1}}{d^0}$ multiplied by a constant λ, serving as the upper bound when setting the shrink rate for each distance. While one could trivially set all shrink rates to zero, doing so fails to control the spatial-temporal consistency, which is detailed in the next section.

To formally justify this relationship, we provide the following theorem based on the pinhole camera model, which establishes a mathematical link between the shrink rate and the resulting change in physical distance.

Theorem 1. *Let d_0 be the initial distance between the drone and the target, and let d_a be the final distance. Let r_a be the target shrink rate under a pinhole camera model with focal length f, and assume the area ratio between the umbrella and the human is a constant $\lambda = \frac{s_u}{s_h}$. If $r_a = \frac{d_a}{\lambda d_0}$, then the drone can be pulled to a distance of d_a, which is shown in Fig. 5.*

Proof. Under a pinhole camera model, the pixel length p of an object of length s at distance d is $p = \frac{fs}{d}$. Initially, the umbrella's pixel length is $p_{u0} = \frac{f\, s_u}{d_0}$. During the attack, the bounding box size is shrunk by a factor r_a, making its pixel length $\frac{f\, s_h}{d_0} r_a$. The drone compensates by advancing until the bounding box size equals the original human pixel length:

$$\frac{f\, s_u}{d}\, r_a \;=\; \frac{f\, s_h}{d_0}.$$

Solving for d yields $d = \lambda r_a\, d_0$. Consequently, if $r_a = \frac{d_a}{\lambda d_0}$, we have $d = d_a$. □

C. Controllable Spatial-Temporal Consistency

Table I summarizes key spatial-temporal consistency defenses proposed to detect adversarial perception attacks. These defenses share a common principle: cross-validating the victim models' predictions against auxiliary estimations derived from independent features or models. Therefore, we design the FlyTrap to be a highly spatial-temporal controllable DPA against ATT drones by introducing the attack target generator (ATG), shown in Fig. 4. ATG enables the attack to explicitly encode both spatial and temporal consistency during optimization, thereby allowing FlyTrap to bypass diverse spatial-temporal defense mechanisms. ATG formulates model-specific constraints to preserve spatial consistency as part of the optimization objective, ensuring intra-model consistency. For instance, we constrain the predicted box to maintain a human-like shape and appear within semantically plausible locations [66], [120]. Additionally, ATG can embed adversarial feature points to mislead feature extractors [114] or craft deceptive human poses to fool pose estimators [71]. These manipulations are feasible due to the attacker's full control over the umbrella pattern in FlyTrap. To ensure inter-model spatial consistency, ATG jointly optimizes multiple perception models, such as SOT, object detector, and pose estimator, such that their outputs align coherently. This coordination ensures spatial consistency within and across perception models.

In addition to spatial alignment, ATG enforces temporal consistency by aligning features across simulated frames. In the simulated images in PDP: $\{\mathbf{I}^0_{\mathbf{A}}, \mathbf{I}^1_{\mathbf{A}}, \ldots, \mathbf{I}^t_{\mathbf{A}}\}$, ATG determines the shrink rates $\{r^0, r^1, \ldots, r^t\}$ for each frame, constrained by the upper bounds specified in Theorem IV-B. By selecting conservative shrink rates, ATG ensures a stable drone trajectory, minimizing abrupt changes that might trigger

TABLE I: Overview of existing representative defense methods leveraging spatial and temporal features to secure perception models in autonomous vehicles. This table excludes sensor-level attacks, as it focuses solely on adversarial attacks targeting machine learning-based perception models. Therefore, sensor attacks (e.g., GPS Spoofing in PhyScout [114]) are not summarized here. OD and MOT refer to Object Detection and Multi-Object Tracking, respectively. ATG represents the attack target generator.

Defense	Victim Model	Attack Goal	Spatial Feature	Temporal Feature	ATG
PercepGuard [66]	OD	Misclassification	Box Shape	Box Behavior, Ego Vehicle States	Inject Box Aspect Ratio
PhyScout [114]	OD	Hiding, Appearing, Misclassification	Box Feature Point	Box Behavior, Ego Vehicle States	Inject Feature Point
VOGUES [71]	MOT	Move-In, Move-Out, Hiding	Component Position	Component Behavior	Inject Human Pose
PhySense [120]	OD, MOT	Misclassification	3D Shape, Texture	Object Behavior, Object Interaction	Inject Human Behavior
VisionGuard [36]	OD	Hiding, Appearing, Misclassification	N/A	Ego Vehicle States	Multi-Stage Shrink Rate

anomaly detectors. The gradual shrink rate design allows DPA to mimic benign scenarios in which the tracked object moves away at a plausible speed. This can prevent sudden box movement [66] or sudden drone movement afterwards [36]. Finally, ATG enforces temporal feature alignment across frames. For instance, it can inject a consistent human pose throughout the PDP simulated attack sequence $\{\mathbf{I}_\mathbf{A}^0, \mathbf{I}_\mathbf{A}^1, ..., \mathbf{I}_\mathbf{A}^t\}$, thus evading defenses that monitor temporal behavior [71], [66], [120].

To demonstrate ATG's generalizability, we categorize existing spatial-temporal consistency defenses into three classes and show how ATG can bypass each class, shown in Table I. For future defense, ATG can also potentially bypass them if they fall within the categorized classes below. (1) *Box feature-based defenses* inspect properties of the bounding box predicted by the victim model (e.g., SOT in our case), such as shape, location, and feature points. Representative examples include PercepGuard [66] and PhyScout [114]. Both approaches examine box-level features. The ATG can set the attack target by manipulating the aspect ratio and feature point within the prediction box accordingly. (2) *Extra visual feature-based defenses* analyze visual features beyond the primary victim model, such as those extracted from additional detectors or pose estimators. Examples include VOGUES [71] and PhySense [120]. These defenses validate the spatial-temporal consistency of auxiliary visual cues, such as human pose and motion behavior, using modules like auxiliary detectors and temporal models. ATG can constrain the inter-model consistency to attack extra models simultaneously to bypass them. (3) *Ego vehicle state-based defenses* detect anomalies by monitoring the smoothness of ego vehicle dynamics, such as velocity and acceleration. VisionGuard [36] is a representative example. However, unlike object detectors studied in VisionGuard, SOT models inherently exhibit spatial-temporal consistency, reducing abrupt vehicle movement changes afterwards [57], [17]. Second, ATG can generate a multi-stage shrink rate to make the drone movement even smoother, thus bypassing defenses relying on vehicle state checking. It should be noted that although PercepGuard also uses vehicle states, it is mainly for assisting the box behavior prediction. PhyScout uses vehicle states for the reconstruction of 3D feature points. Neither of them uses ego states as major or direct evidence for detecting the underlying perception attacks. Therefore, we don't include them in this category.

D. Overall Optimization Pipeline

After introducing FlyTrap's key novel designs, we present the comprehensive optimization process from data collection to real-world robustness in this section, as shown in Fig. 4.

1) Dataset Construction: We first introduce the dataset construction process to train the adversarial pattern. Specifically, we collect aerial videos that depict common deployment areas for ATT drones. Each video is split into two segments, the template video and the search video: $V = \{V_{\text{tplt}}, V_{\text{search}}\}$. The first segment tracks a person, corresponding template videos $V_{\text{tplt}} = \{\mathbf{I}_{\text{tplt}_1}, \mathbf{I}_{\text{tplt}_2}, ...\}$, where $\mathbf{I}_{\text{tplt}_n} \in \mathbb{R}^{H\times W\times 3}$ represent n-th template frame in the video. The template frames are used to initialize the tracker. The second segment records the same subject deliberately opening an umbrella and pointing it at the drone to simulate adversarial behavior, resulting in search frames $V_{\text{search}} = \{\mathbf{I}_{\text{search}_1}, \mathbf{I}_{\text{search}_2}, ...\}$. $\mathbf{I}_{\text{search}_n} \in \mathbb{R}^{H\times W\times 3}$ represents the n-th search frame, where the tracker makes predictions. These search frames serve as inputs to the PDP (Section IV-B), which simulates adversarial umbrellas and distance-pulling dynamics. We further perform automated labeling and down-sampling to pre-process the dataset.

2) Adversarial Objective Function: We leverage the attack target from the ATG as our optimization goal. Specifically, for each PDP time step t, we guide the SOT model to predict a bounding box of size w_a^t and h_a^t and centered at cx_a^t and cy_a^t, represented as a tuple $\mathcal{P}_a^t = (cx_a^t, cy_a^t, w_a^t, h_a^t)$. We set all cx_a^t and cy_a^t to the center of the umbrella by default:

$$\mathcal{L}_{\text{loc}} = \frac{1}{NMT}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{T}\left\|\mathcal{P}_{i,j}^t \ominus \mathcal{P}_a^t\right\|, \tag{3}$$

where N, M, and T are the overall number of search video frames, tracking candidate proposals, and PDP time steps. Additionally, the control algorithms are designed to react to tracking results only if their confidence scores are sufficiently high to ensure safe autonomous flight [23]. Thus, we maximize the predicted confidence $score_i$ to ensure that our injected tracking results can propagate throughout the ATT drones:

$$\mathcal{L}_{\text{cls}} = \frac{1}{NMT}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{T}\left[-\log(score_{i,j}^t)\right]. \tag{4}$$

Beyond SOT, we also co-optimize across multiple models to satisfy spatial-temporal consistency constraints for adaptive attacks. For example, to achieve spatial consistency, we assign the same location and confidence objectives (Eq. 3 and 4) to an object detector. For those defenses that use auxiliary pose

estimation model [71], we assume the attackers can control their pose right before launching the attack, and preserve the temporally consistent pose by injecting it in consecutive attack frames. Specifically, we optimize the pose estimation heat map $\mathbf{H}^t$ at each time step to remain close enough to a benign reference map $\mathbf{H}_{\text{benign}}$, which is averaged across the last few frames in the template video before umbrella deployment:

$$\mathcal{L}_{pose} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left\| \mathbf{H}_i^t - \mathbf{H}_{\text{benign}} \right\|. \tag{5}$$

Lastly, to maintain physical-world realizability, we regularize the adversarial patterns with a total variation (TV) loss:

$$\mathcal{L}_{\text{TV}}(\mathbf{A}) = \sum_{i=1}^{H_a-1} \sum_{j=1}^{W_a-1} \|\mathbf{A}_{i+1,j} - \mathbf{A}_{i,j}\| + \|\mathbf{A}_{i,j+1} - \mathbf{A}_{i,j}\|, \tag{6}$$

where $\mathbf{A}_{i,j}$ represent pixel values at location (i, j) on the adversarial pattern $\mathbf{A}$ before rendering. Finally, we optimize the adversarial pattern as a weighted sum of all the objectives:

$$\min_{\mathbf{A}} \mathbb{E}_{\mathcal{T} \sim \mathcal{T}_{\text{compose}}} \big[\sum_{k} w_k \mathcal{L}_k \big], \tag{7}$$

where w_k is the tuned weight to balance the k-th objective function and $\mathcal{T}$ denotes the transformation detailed below.

3) Physical-world robustness: To overcome the influence of innumerable physical factors, we stack a set of expectations over transformation (EoT) within the optimization process [2]. In the rendering operation, we randomly select the camera elevation angle $\mathcal{T}_1(\cdot) : \theta \sim [-5°, 5°]$ and azimuth angle $\mathcal{T}_2(\cdot) : \phi \sim [-5°, 5°]$ to simulate the attacker pointing the umbrella slightly off the camera center. We randomly sample the angle of rotation $\mathcal{T}_3(\cdot) : \psi \sim [-20°, 20°]$ of the umbrella to simulate the imperfect vertical direction of the physical adversarial pattern. Additionally, we add image transformations to the adversarial patterns, including Gaussian noise ($\mathcal{T}_4$), brightness ($\mathcal{T}_5$), contrast ($\mathcal{T}_6$), saturation ($\mathcal{T}_7$), and hue transformation ($\mathcal{T}_8$) to simulate complex physical world environments. The final transformation is composed of all the transformations $\mathcal{T}_{\text{compose}} = \{\mathcal{T}_1 \circ \mathcal{T}_2 \circ \cdots \circ \mathcal{T}_8\}$. Additionally, the PDP naturally incorporates estimation error due to imperfect physical modeling, accounting for the real-world imperfection control assumed in the closed-loop simulation.

V. Attack Evaluation

A. General Experimental Setups

1) Dataset collection: We collected an aerial-view dataset for training and evaluation. The dataset includes video recordings featuring four individuals with diverse appearances and covers four typical drone deployment environment types, including two grass fields, two parking lots, one bare ground area, and one drivable road. For each combination, we recorded two videos: one for training and one for testing. In total, the dataset includes 23 training videos comprising 11,898 frames and 25 evaluation videos comprising 13,594 frames. Ethics considerations can be found in Section VIII.

2) Models: In our experiments, we choose SiamRPN-based [57] SOT models as victims, following prior work [69], due to their strong trade-off between tracking accuracy and computational efficiency. To further broaden our evaluation, we also include MixFormer, a state-of-the-art Transformer-based SOT model [103], which represents the recent trend toward more expressive yet computationally intensive tracking architectures. By incorporating both CNN-based and Transformer-based models, this combination provides comprehensive coverage of the current SOT model landscape.

3) Metrics: Under the DPA setting, we define evaluation metrics to capture the system-level impact of the attacks. Specifically, we define two metrics: (1) open-loop attack success rate (ASR_{open}) and (2) closed-loop attack success rate ($\text{ASR}_{\text{closed}}$). ASR_{open} is defined as successful if all of the following conditions are satisfied: (1) to ensures the drone is expected to be pulled to within the attacker-desired distance: the bounding box area must be smaller than a shrinkage threshold r_a of the umbrella areas; (2) to ensures the ATT system doesn't lose track and thus fail in distance-pulling: the prediction confidence must exceed a predefined threshold $score_a$ and (3) to ensure the umbrella is the trigger: the attacked prediction bounding box must lie entirely within the umbrella boundary:

$$\mathcal{C}_{\text{open}} : \begin{cases} a \leq r_a \cdot a_u, \\ score \geq score_a, \\ (c_x, c_y, w, h) \subseteq (c_{u_x}, c_{u_y}, w_u, h_u), \end{cases}$$

where a denotes the bounding box area and the u subscript denotes umbrella. We evaluate frames from the testing dataset and compute the ASR_{open} as:

$$\text{ASR}_{\text{open}} = \frac{\sum_{i=1}^{N} \mathbb{I}(\mathcal{C}_i)}{N},$$

where $\mathbb{I}$ is the indicator function, $\mathcal{C}_i$ represents the condition for the i-th sample, and N is the total number of frames. To capture ASR comprehensively across varying threshold settings, we introduce a metric similar to mean Average Precision (mAP) used in object detection [27], [59]. We define mean ASR_{open} ($\text{mASR}_{\text{open}}$) as the average ASR_{open} over a set of thresholds r_a and $score_a$ ranging from 0.1 to 0.9 in increments of 0.1 to provide broad coverage. For the $\text{ASR}_{\text{closed}}$, we define success if the final distance d between the drone and the attacker is below a distance threshold d_a:

$$\mathcal{C}_{\text{closed}} : d \leq d_a.$$

The $\text{ASR}_{\text{closed}}$ is computed as the average success rate across multiple real-world drone flights. The success criterion for $\text{ASR}_{\text{closed}}$ is straightforward: the distance threshold can be the maximum range to capture the drone (e.g., 9 meters [19]) or it can be the working distance for sensor attacks (e.g., 6 meters for projector [124]) or hitting distance (e.g., 0.5 meters).

B. Attack Effectiveness

1) Evaluation Methodology: As a baseline, we use target photos (TGT) cropped from the first frame of each training

TABLE II: Evaluation of attack effectiveness ($mASR_{open}$). SiamAlex, SiamRes, and SiamMob refer to SiamRPN [57] combined with AlexNet [54], ResNet [37], and MobileNet [41], respectively. The FlyTrap attack consistently outperforms the TGT baseline, despite its visual similarity to the tracked object. $FlyTrap_{PDP}$ consistently outperforms the vanilla version.

Attack	MixFormer	Siam-Alex.	Siam-Res.	Siam-Mob.	Avg.
TGT	46.3%	37.2%	24.9%	35.5%	36.0%
FlyTrap	42.0%	17.0%	44.3%	32.1%	33.9%
$FlyTrap_{PDP}$	78.7%	35.6%	50.8%	49.1%	53.6%

TABLE III: Evaluation of scenario universality for attacks across unseen target-location combinations ($mASR_{open}$). We evaluate the attack on two unseen people and two unseen locations with different target-location combinations. The FlyTrap in this table is the version with the PDP design.

	Scenario Universality					
	Location (6 Videos)		Person (7 Videos)		Both (6 Videos)	
Model	TGT	FlyTrap	TGT	FlyTrap	TGT	FlyTrap
MixFormer	25.5%	85.9%	11.6%	40.4%	6.8%	34.1%
SiamRPN-Alex	34.3%	50.2%	24.2%	67.9%	21.7%	33.0%
SiamRPN-Res	20.7%	55.2%	10.4%	63.5%	9.9%	42.8%
SiamRPN-Mob	28.8%	55.9%	13.8%	54.5%	12.2%	26.0%
Average	27.3%	61.8%	15.0%	56.6%	12.6%	34.0%

video, corresponding to the same person and location, as they naturally resemble the genuine target being tracked. The TGT can be regarded as a simple human figure printing baseline attack. We use grid search to find the ratio of the printed human figure to the umbrella that can maximize $mASR_{open}$ and use that for fair baseline comparisons. More TGT generation details are included in the Appendix A. Since TGT also applies an image on the umbrella surface, the $mASR_{open}$ can be naturally applied to it. This evaluation involved 4 people $\times$ 4 locations = 16 TGT combinations in total. The $mASR_{open}$ was averaged over 16 TGTs $\times$ 6 testing videos = 96 combinations of experiments. Regarding FlyTrap, we select two people as the tracked target and two locations as the background for training. Then, we evaluate FlyTrap on the 6 testing videos of the same target person and background. Vanilla FlyTrap can also be considered as a baseline from the previous SOT shrinking attack [20], [115], but with our new contributions of the umbrella modeling, DPA-specific objective function design, and attack vector-specific robustness design.

2) Experiment Results: The main results of our evaluation are presented in Table II. We find TGT, even though it visually matches the genuine person being tracked, performs considerably limited, with an average $mASR_{open}$ of 36.0% across all victim models. In comparison, $FlyTrap_{PDP}$ achieves a much higher $mASR_{open}$ of 53.6% on average, underscoring its effectiveness. The comparison between FlyTrap with and without PDP design shows its effectiveness in further shrinking the area, which is also observed in the physical experiments. We also study the robustness of FlyTrap to environmental distractions, where multiple similar but unobstructed objects (e.g., other passersby) appear in the same scenario and find that FlyTrap can cause consistent attack effects given the presence of other visual distractions. Please refer to our website [113] for more details and demonstrations.

It's worth noting that $mASR_{open}$ is a challenging metric. Specifically, assume the $\mathcal{C}_{\text{open}}$ can always be satisfied when $\forall\ r_a \geq 0.5,\ \mathbb{I}(\mathcal{C}_{\text{open}}) = 1$ and $\forall\ r_a < 0.5,\ \mathbb{I}(\mathcal{C}_{\text{open}}) = 0$, the $mASR_{open}$ will be 50.0%. However, this can already shorten the tracking distance to half of its original distance as indicated by Theorem IV-B. We show in physical experiments (Section V-F2) that the shrink rate can continuously decrease as the distance decreases. Therefore, the $mASR_{open}$ primarily serves for digital, scalable evaluation before printing adversarial patterns for physical evaluation. Thus, even though the absolute number of $mASR_{open}$ might not seem as high as expected, we find it already sufficient enough to cause closed-loop impacts as indicated by our physical closed-loop experiments (Section V-F5).

C. Scenario Universality

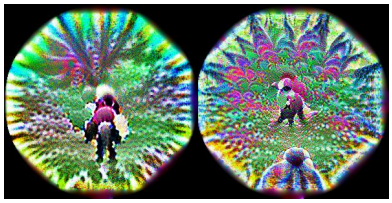
1) Evaluation Methodology: For TGT, we apply the same set of images from Section V-B to videos of unseen scenarios, including different target persons and/or different background locations. The results are derived from 16 TGTs $\times$ 19 testing videos = 304 combinations of tests. For FlyTrap, we use the same set of adversarial patterns optimized in Section V-B to an unseen person and/or unseen background. We report the $mASR_{open}$ of universality to location (6 testing videos), to person (7 testing videos), and both (6 testing videos).

2) Experiment Results: In Table III, we observe that the TGT shows limited universality, even for the same tracked person with a different background. Its universality to location is only 27.3% across all models. The universality to person and to both are even worse. Therefore, TGT might only be useful if the attacker knows the exact scenario, including both the person and location. On the contrary, FlyTrap shows a significantly better universality of 61.8%. The universality to location and to person can achieve comparable $mASR_{open}$ as effectiveness shown in Table II, suggesting that FlyTrap, when trained on a subset of location or person, can generalize effectively to unseen individuals or environments, satisfying the need to attack ATT drones in unknown deployment places. However, when both the person and location are unseen, the $mASR_{open}$ are slightly lower, but still 21.4% higher than TGT.

D. Attack Transferability

1) Evaluation Methodology: We employ the FlyTrap optimized from one victim SOT model for transferring to attack another. We consider FlyTrap with and without PDP designs. We study the transferability of FlyTrap without PDP as we observe an interesting adversarial pattern: the human-shape pattern in Fig. 6 (a), which might be more transferable as it visually resembles a standing human. Then, we report the $mASR_{open}$ on the same set of testing videos as Section V-B1.

2) Experiment Results: The main results are shown in Fig. 7. Notably, we observe that the human-shape patterns indeed have better transferability, with an average of $mASR_{open}$



(a) (b)

Fig. 6: Visualization of adversarial patch designs against MixFormer [17]. (a) Umbrella pattern without progressive distance-pulling, which achieves high transferability due to its visual resemblance to a standing human. (b) Umbrella pattern with progressive distance-pulling, exhibiting a structured cascade pattern that enhances continuous distance-pulling effects.

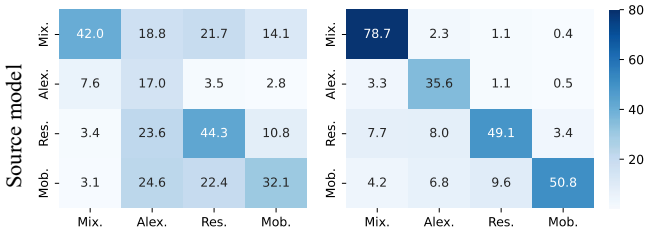


Fig. 7: Attack transferability evaluation ($mASR_{open}$%). (a) FlyTrap pattern optimized without progressive distance-pulling (PDP) design; and (b) with PDP design.

of 13.1% compared to FlyTrap with PDP design, which is 4.0%. The phenomenon can be explained by the visual appearance of the optimized pattern shown in Fig. 6. Specifically, Fig. 6 (a) shows visual resemblance to a standing human, which potentially benefits the transferability, while Fig. 6 (b) exhibits a structured cascade pattern that enhances continuous distance-pulling effects but is more model-specific. Among them, the adversarial pattern against MixFormer [17] shows the highest transferability of 18.2% $mASR_{open}$ on average. Such a level of transferability can already achieve effective DPA against black-box commercial systems (Section V-G). The results suggest a trade-off between a more continuous distance-pulling attack to a more transferable attack, which we acknowledge as a limitation in Section VI-B.

E. Spatial-temporal Consistency

We select one defense for each of our three categorized classes (Section IV-C) to evaluate FlyTrap's spatial-temporal consistency for each defense type, using the same set of testing videos in effectiveness evaluation (Section V-B).

1) PercepGuard Evaluation: We adopt the released behavior LSTM [38] model in the official codebase [65]. The input to the LSTM model is the tracked bounding box prediction in the last ten frames. The output of the LSTM model is a probability distribution over several classes. We regard the alarm as raised if the prediction is not "pedestrians". The results are shown in Table IV. The original PercepGuard true alarm rate (TAR) in benign case and false alarm rate (FAR) in attack case are 99.0% and 5.0%, respectively [66], [71]. Our evaluation found a similar trend of low FAR in benign tracking cases and high TAR for vanilla FlyTrap. However, with our ATG design, the $FlyTrap_{ATG}$ attack can decrease the TAR to 2.8%, even lower than the FAR of around 5.0% reported in the original paper [66]. The FAR in our evaluation is lower because our collected dataset is single object tracking scenarios, while in the original evaluation, it's tested in driving scenarios with a more complex environment (i.e., the BDD dataset [119]), thus leading to slightly higher FAR.

TABLE IV: Evaluation of the PercepGuard [66] defense. We report False Alarm Rates (FAR) under benign inputs and True Alarm Rates (TAR) under vanilla and $FlyTrap_{ATG}$ attacks. Notably, the $FlyTrap_{ATG}$ achieves an average detection rate of only 2.8%, which is lower than the 5% benign PercepGuard FAR reported in [71].

Model	**False Alarm Rate** Benign	**True Alarm Rate** FlyTrap	FlyTrap$_{ATG}$
MixFormer	2.6%	78.0%	2.2%
Siam-Alex	0.0%	41.2%	4.9%
Siam-Res	0.0%	55.2%	0.0%
Siam-Mob	0.0%	68.2%	4.2%
Average	0.7%	60.7%	2.8%

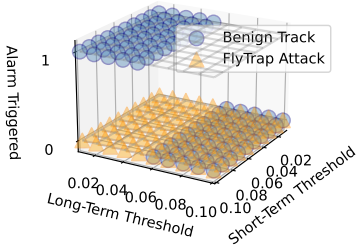


Fig. 8: Evaluation results of VisionGuard [36]. In the z-axis, 0 represents alarm is not activated, and 1 represents that the alarm is triggered. FlyTrap can bypass it across all the tuned thresholds, even when the benign track triggers the alarm.

2) VOGUES Evaluation: VOGUES [71] was originally proposed for defense Multiple Object Tracking (MOT). Therefore, we follow the original setups while making necessary adaptations for SOT. Following their setups [70], we adopt YOLOv3 [85] as the object detector and AlphaPose [29] as the pose estimator. We compute the spatial consistency by choosing the object detection prediction that has the highest Intersection of Union (IoU) with the SOT prediction to avoid high false positive rates during the SOT adaptation. Since VOGUES doesn't release the LSTM model they use to evaluate the consistency of the human pose, we reproduce it by following their setups: we train the LSTM model using the pose from the UCF101 dataset [99]. Same to their setups, we set the OD and OT prediction IoU threshold as 0.5 and the LSTM threshold as 0.5. If any of the values is below the threshold, an alarm will be raised. We observe that SOT can mostly operate normally even if a benign umbrella is used as camouflage. Therefore, the defense is desired to tolerate the spatial-temporal anomaly induced by a normal umbrella,

TABLE V: VOGUES [71] defense evaluation. $\text{FlyTrap}_{\text{ATG}}$ can largely decrease the alarm rate across models and almost achieve consistently lower alarm rates than a benign umbrella.

Model	False Alarm Rate		True Alarm Rate	
	Benign	Umbrella	FlyTrap	$\text{FlyTrap}_{\text{ATG}}$
MixFormer	3.1%	51.6%	93.7%	32.4%
Siam-Alex	4.2%	75.6%	89.3%	77.9%
Siam-Res	3.4%	73.0%	91.4%	54.4%
Siam-Mob	3.9%	65.1%	80.8%	44.8%
Average	3.7%	66.3%	88.8%	52.4%

avoiding an unnecessarily high false alarm rate that hampers ATT drones from operating normally in this case.

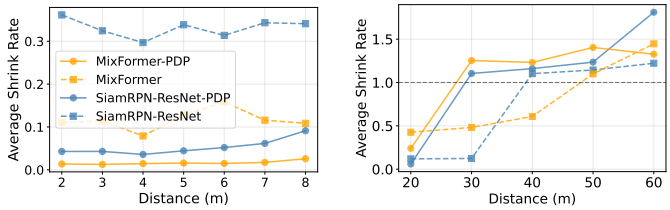
Table V shows that with the ATG design, $\text{FlyTrap}_{\text{ATG}}$ can largely decrease the alarm rate across models and almost achieve consistently lower alarm rates than a benign umbrella. We find that the relatively high alarm rate compared with benign tracking is caused by imperfect universal attacks against the object detector across multiple frames and videos, which is an observed limitation in existing attacks against object detectors [36]. Nonetheless, the ablation study of $\text{FlyTrap}_{\text{ATG}}$ is already sufficient in justifying the effectiveness of our ATG design and is significantly lower than the effectiveness level that the original VOGUES used to claim success (98.4%).

3) VisionGuard Evaluation: We use the official implementation for evaluation [35]. To collect drone states, we use the Clover drone simulator [28], which uses Gazebo [53] as the backend and PX4 [83] as the firmware. We simulate the benign tracking and the FlyTrap attacked tracking in `OFFBOARD` mode [83] by publishing to the ROS topic `mavros/setpoint_position/local` to set the target point at a frequency of 20 Hz, simulating the SOT model's inference FPS. Following their setup, we only consider the drone and person moving along the x-axis for simplicity, without losing generality. The drone's states can be retrieved by subscribing to `mavros/local_position/velocity_local`, which is used as the input to the ARIMA [7] states estimation model. Finally, we use one benign sequence to train the model for predicting the drones' states and test on the FlyTrap attack sequence and another benign track sequence. Each sequence has around 130 frames sampled every 0.1 seconds. We iterate the alarm threshold, including long-term residual and short-term residual, and set the accumulated threshold to 2. In Fig. 8, we find that FlyTrap can bypass VisionGuard [36] across all the tuned parameters, even when the benign track triggers the alarm. The results suggest that the fundamental rationale of VisionGuard to aim for inconsistent attack effects in object detection has limited applicability to the ATT drone context, where the SOT prediction is temporally consistent by nature.

4) Impact on Attack Effectiveness: In Table VI, we evaluate the impact of ATG design on attack effectiveness. The results show that the ATG design to constrain spatial-temporal consistency has a subtle impact (within 10%) on the attack performance, which is still significantly higher than TGT (in Table II). For example, FlyTrap against MixFormer with spatial-temporal constraints can achieve similar $\text{mASR}_{\text{open}}$ compared with FlyTrap without constraints: 76.8% for PercepGuard [66] and 69.4% for VOGUES [71]. Interestingly, we find the FlyTrap with ATG can even significantly boost the $\text{mASR}_{\text{open}}$ for SiamRPN-AlexNet to 51.1% and 41.4%.

TABLE VI: Attack effectiveness evaluation ($\text{mASR}_{\text{open}}$) with spatial-temporal constraint tailored for different defenses. The constraint has a subtle impact (within 10%) on the attack performance across the models, which are still significantly higher than TGT.

ATG	MixFormer	Siam-Alex.	Siam-Res.	Siam-Mob.
-	78.7%	35.6%	50.8%	49.1%
PercepGuard	76.8%	51.1%	53.5%	47.6%
VOGUES	69.4%	41.4%	40.6%	40.5%



(a) Short-range evaluation (b) Long-range evaluation

Fig. 9: Physical evaluation of average shrink rate versus distance. (a) $\text{FlyTrap}_{\text{PDP}}$ achieves a lower shrink rate across all distances in short-range evaluation since the shrink rate continuously decreases as distance decreases. (b) FlyTrap works well below 20m and can potentially extend to 30m or 40m. *PDP* means using PDP during attack optimization.

F. Physical-World Attack Evaluation

1) Open-Loop Evaluation Setups: We create real-world adversarial umbrella prototypes by uploading the optimized adversarial patterns to an online umbrella-printing service. We record 10-second videos at varying distances using a 4K resolution smartphone camera (iPhone 16). This results in around 600 frames with a resolution of 3840 $\times$ 2160 for each video. Then, we run the SOT model offline and evaluate the shrink rate. We report the average shrink rate recorded at different distances. We use umbrellas printed with FlyTrap patterns optimized against MixFormer and SiamRPN-ResNet, both with and without PDP design. For both, the target shrink rate in the objective function (in Section IV-D2) is set to 0 to study the effects of closed-loop distance-pulling effects.

2) Open-Loop Experiment Results: We study the average shrink rates at different distances, shown in Fig. 9. In the short-range evaluation, for both SiamRPN and MixFormer, the average shrink rate of $\text{FlyTrap}_{\text{PDP}}$ decreases as the distance decreases since the higher resolution on the adversarial pattern can cause an even lower shrink rate. On the other hand, the shrink rate of the vanilla FlyTrap remains almost the same since it's locked onto one fixed area in the pattern (e.g., the human shape area in Fig. 6 (a)). The results suggest that PDP enables finer optimization at higher resolution, resulting in a

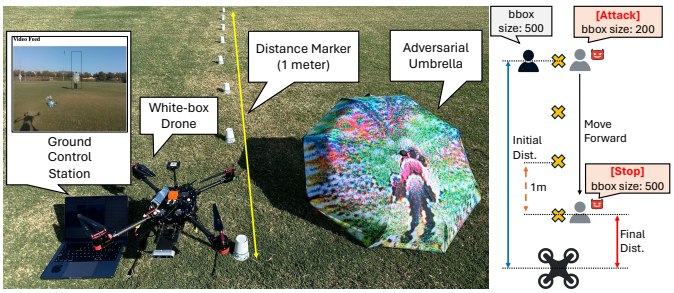


Fig. 10: Physical closed-loop evaluation setups. An operator carries the drone to move forward until the shrunk box size matches the original size.

significantly smaller tracked area. We further evaluate at long-range distances to study the maximum working distance for the FlyTrap attack. Optimized using collected videos under 20 meters, we find FlyTrap works well at that distance and can potentially extend to 30 or 40 meters. The distance is beyond the maximum functional distance of available ATT drones with 4K videos, which are typically around 20 meters [25], [26], [81]. It should be noted that the training dataset we collect mainly includes close-range footage. Thus, FlyTrap$_{\text{PDP}}$ might not be well-optimized for the long-range attack case. We leave it as future work to further study the long-range attack capabilities. More discussions are in the Appendix B.

3) Closed-Loop Evaluation Setups: To evaluate our attack in physical, closed-loop setups, we build our experimental platform using Hylobro X500 v2 drone [39], a medium-lift quadcopter powered by a Pixhawk flight controller. The system uses Robot Operating System (ROS) [84] as the communication backbone. We detail the implementation in Appendix C.

4) Closed-loop Evaluation Methodology: We conduct experiments with the same location and target person as our training videos (Section V-B). For safety and distance measurement issues, we simulate ATT behavior without a physical takeoff by manually maneuvering the drone on the ground. The experimenter, who acts as the attacker, then initiates the FlyTrap attack and manually moves forward until the tracking box size matches the initialization (shown in Fig. 10), which closely approximates the outcome of an autonomous closed-loop flight. We test all four models with FlyTrap initialized from 7 various starting distances from 6 to 12 meters. For closed-loop attack success criterion d_a (Section V-A3), we choose each of the three attack goals in Section II-B: *A1*: $d_a = 9$ for net gun capturing [73], [74], [100]; *A2*: $d_a = 6$ for binocular camera projection attack [124], as it directly related to drone sensor spoofing attacks with range limits; and *A3*: $d_a = 0.5$ for crashing, as it within human arm length to push the umbrella. Finally, we report the resulting ASR$_{\text{closed}}$.

5) Closed-loop Experiment Results: As shown in Table VII, FlyTrap$_{\text{PDP}}$ substantially reduces the tracking distance of the ATT system to be within the range of capturing, sensor attacks, or direct crashes. These results confirm the physical feasibility of our threat model and demonstrate that the proposed PDP can significantly affect the tracking distance of autonomous tracking drones in closed-loop control. As shown in Fig. 11, since we use RealSense camera [43], which has lower resolution of 640×480 compared to the videos captured by the smartphone in Section V-F1, the phenomenon that the shrink rate decreases as the distance decreases is even more obvious compared to those in Fig. 9a. The results highlight the physical-world impact and FlyTrap$_{\text{PDP}}$'s capability to progressively pull the drone as it approaches.

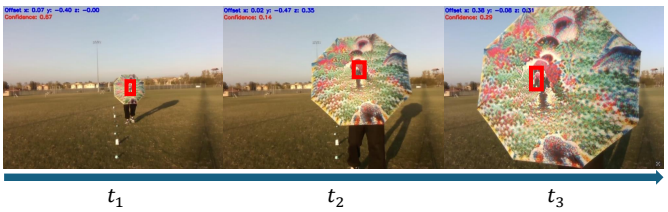


Fig. 11: Physical evaluation of FlyTrap$_{\text{PDP}}$ against MixFormer [17]. The video is captured by an on-drone RealSense camera with a resolution of 640×480. The shrink rate decreases largely as the distance decreases, enabling the progressive distance-pulling effects. Bold the box for clarity.

TABLE VII: White-box physical closed-loop evaluation results of ASR$_{\text{closed}}$ against different models under different attack goals. *w/ PDP* means using PDP during attack optimization.

Victim Model	Capture (9 m)	DoubleStar (6 m)	Crash (0.5 m)
MixFormer	100.0%	100.0%	0.0%
MixFormer w/ PDP	100.0%	100.0%	**100.0%**
Siam-Alex	100.0%	100.0%	0.0%
Siam-Res	100.0%	100.0%	0.0%
Siam-Res w/ PDP	100.0%	100.0%	**100.0%**
Siam-Mob	100.0%	85.7%	0.0%

(a) (b)

Fig. 12: Commercial evaluation. (a) Crafted FlyTrap physical umbrellas and three commercial drones. (b) Net gun equipment set used for an end-to-end attack demonstration.

G. Commercial System Attack Evaluation

To assess DPA's real-world viability and highlight potential vulnerabilities in widely accessible commercial products, we conduct a black-box evaluation on three consumer-grade drones equipped with visual tracking systems. As noticed in [106], the obscure implementations in commercial systems can heavily undermine the phenomenon observed in academia prototypes. Therefore, we evaluate whether our proposed DPA can be conducted in the physical world against real-world products instead of testing our PDP and ATG design, which is specifically for white-box systems (Section II-C).

1) Evaluation Setups: We acquire three commercially available drone products, including DJI Mini 4 Pro [22], DJI

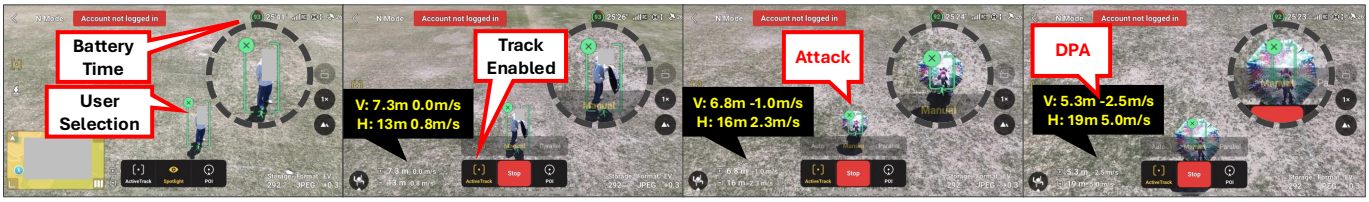


Fig. 13: Screenshots from the DJI Mini 4 Pro remote controller interface during a real-world FlyTrap DPA. The green tracking box indicates the output of DJI's built-in tracking model (zoomed-in view in the figure with a grey dotted border for clarity). In the third frame, the tracker erroneously locks onto a small subregion within the adversarial umbrella pattern. We highlight the vertical and horizontal position and their corresponding velocity of the drone. In the fourth frame, the altitude decreases to 5.3 meters at -2.5 m/s while the drone is moving forward at 5 m/s, suggesting the drone rapidly descends toward the attacker.

TABLE VIII: Commercial evaluation of $\text{ASR}_{\text{closed}}$. Distance limitation for each attack is used as the threshold for $\text{ASR}_{\text{closed}}$. A human-shaped FlyTrap umbrella successfully deceives three consumer drones, with crash outcomes on DJI Neo and HoverAir. N/A means the drones' preset initial tracking distance is already within the distance. These findings demonstrate the real-world existence of the ATT vulnerabilities we identified.

Attacks	**DJI Mini 4 Pro**	**DJI Neo**	**HoverAir**
Capturing (9 m)	60.0%	N/A	N/A
DoubleStar (6 m)	30.0%	N/A	N/A
Crash (0.5 m)	0.0%	60.0%	80.0%

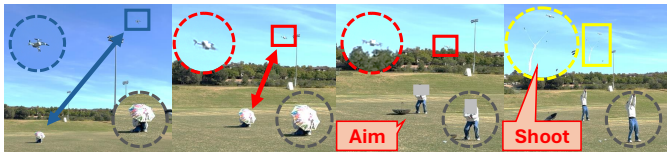


Fig. 14: End-to-end FlyTrap-enabled DPA demonstration: *A1: drone capturing*. We use a net gun to shoot the DJI Mini 4 Pro with *Active Tracking* feature when the tracking distance is pulling closer to the attacker. Zoomed-in view in the figure with dotted borders for clarity.

Neo [24], and HoverAir [40]. We employ the same umbrella design described in Section V-F3, shown in Fig. 12 (a). We purchase a net gun [74] to demonstrate the DPA-enabled drone capturing attacks, shown in Fig. 12 (b).

2) Evaluation Methodology: We place ground markers at fixed intervals (Δd) and have an observer walk parallel to the drone's flight path to estimate displacement. The flying altitude h is retrieved from the flight log or viewed directly on the controller interface. The distance between the drone and the person is derived by $d = \sqrt{h^2 + (n\Delta d)^2}$. We record the final distance under attacks and report the resulting $\text{ASR}_{\text{closed}}$. Each drone is tested in 10 separate flights with fixed initial distances. For the DJI Mini 4 Pro, we set the initial distance as 12 meters, and for the DJI Neo and HoverAir drones, the tracking distance is factory set to around 2 meters. For each flight, we begin by powering up the drone, taking off, and then activating the tracking. We don't report capturing and DoubleStar [124] attack for DJI Neo and HoverAir since their preset initial tracking distance is already within those ranges.

3) Experiment Results: We find that one of the umbrellas, i.e., adversarial pattern against MixFormer shown in Fig. 6 (a), can successfully deceive all three tested drones. As shown in Table VIII, we find that with FlyTrap, we can capture or launch sensor attacks on the DJI Mini 4 Pro with a success rate of 60% and 30%, respectively. In Fig. 13, we record the screen of the DJI remote controller to verify that the distance-pulling is indeed caused by a shrunk box instead of other factors. In Fig. 14, we show an end-to-end DPA-enabled drone-capturing attack. After conducting the DPA, the tracking distance is shortened largely, allowing the attacker to aim and then shoot the capturing net against the drone. For DJI Neo and HoverAir, we observe that these two ultra-light drones are easy to crash, with a success rate of 60% and 80%, respectively. We suspect they lack the tracking state-verification mechanisms found in more advanced models like the DJI Mini 4 Pro [23]. Thus, they try to catch up with the "shrunk" object at a relatively high speed, causing the observed collision. Fig. 15 showcases an end-to-end DPA-enabled crashing attack on the HoverAir drone. The attacker can physically hit the drone using the umbrella to cause a collision and crash. We also empirically find that crouching down to cover the whole body can increase the attack success rate, as shown in Fig. 13 and 14.

The failure of other umbrellas might be due to the white-box model, which they are optimized against, being convolutional-based models. Ma et al. find that adversarial examples generated using Transformer-based models tend to be more transferable than CNN-based models [63]. We also find that adversarial patches against SiamRPN, which is a CNN-based SOT model, show more model-specific patterns (second to fourth umbrellas in Fig. 12 (a)) compared to the human-shape pattern from MixFormer (umbrella in Fig. 6).

Additionally, we find the black-box commercial results consistent with transferability experiments (Section V-D): the pattern with a human-like shape, with the highest transferability among others (18.2%), can potentially transfer to commercial drones. Despite only one of our crafted umbrellas succeeding, we reveal the first demonstration of the proposed DPA vulnerabilities in deployed commercial drone systems, thus justifying the security problem we identified, and demonstrate two out of three use cases of DPA (*A1* and *A3* in Section II-B). We have already performed responsible vulnerability disclosure to

Fig. 15: End-to-end FlyTrap-enabled DPA demonstration: *A3: drone crashing*. We use the umbrella to hit the HoverAir drone with *Dolly Track* feature when the tracking distance is pulled within the hitting distance to the attacker.

the corresponding manufacturers (Section VIII).

H. Attack Stealthiness Evaluation

The proposed FlyTrap attack is stealthy since the umbrella is folded for most of the carrying time. In this section, we conduct a user study to further justify the stealthiness during the deployment time.

1) Evaluation Setups: The survey consists of two parts. In the first part, we focus on the attack vector, the umbrella: we investigate if it's uncommon to use an umbrella on a non-rainy day. Next, given the adversarial pattern, we investigate if people feel suspicious about it. We released the survey on the Prolific platform [82] with 200 participants sampled to reflect a broad demographic distribution in the United States. For the user study, we go through the IRB review process of our institution and receive confirmation of the self-exempt subject search categorization. More details can be found in the ethics discussion in Section VIII.

2) Experiment Results: In terms of the usage of umbrellas, we find 78.6% of the participants think it's not abnormal to them when seeing someone using an umbrella on a non-rainy day. In terms of the adversarial patterns, some portion of the participants might think our FlyTrap pattern is eye-catching (∼13%), and rank 3rd among all the candidate umbrellas. However, only 5.9% of participants found the FlyTrap pattern suspicious, lower than the percentage of 26.7% who selected "*none of the umbrellas are suspicious*". The results suggest the FlyTrap is a deployable attack in the real world without being considered suspicious by the general public, even during the launch time. Please refer to our website [113] for full results.

VI. Discussion and Limitations

A. Other countermeasures

In addition to spatial-temporal consistency checking, there are model-level defense strategies such as certified robustness [55], [110], [109], [111], adversarial training [96], [33], [64], and input transformation [34], [112]. Certified defenses provide theoretical guarantees of model robustness against white-box attacks, typically by ensuring bounded prediction error in the presence of adversarial perturbations. However, existing certified robustness techniques and adversarial training methods primarily target misclassification attacks, making their direct application to SOT models non-trivial due to the fundamentally different attack objectives and model behaviors. To the best of our knowledge, there is no specific certified robustness or adversarial training for SOT defense. Input transformation-based defenses, in contrast, are ineffective under strong Expectation over Transformation (EoT) optimization [2]. Jia et al. [45] proposed a defense specifically tailored for SOT models. However, their method targets pixel-level perturbations and relies on run-time gradient-based iterative denoising. The gradient back-propagation process is well-known to be much slower than the inference, which is computationally intensive and thus impractical for real-time ATT drone applications [68]. Specifically, ATT drones are more likely to lose target if the inference speed is limited, given the temporal dependency of their working mechanisms. Therefore, future defense needs to improve the efficiency to support real-time ATT applications.

B. Limitations

We acknowledge the limitation of FlyTrap's black-box transferability. Our approach is mainly designed to achieve better attack effects for white-box ATT systems, and thus might sacrifice the black-box transferability. However, it should not be the major concern given (1) the existing advancement in reverse engineering (in Section II-C) and (2) the existing approach to boost black-box transferability in attacking SOT [118], [46]. They are applicable to FlyTrap by replacing the gradient-based optimization with their black-box searching, which is out of the scope of this work. Furthermore, the black-box commercial system implementation can influence how the commercial systems react towards the attack [106]. Nonetheless, we've shown the real-world existence of the vulnerabilities by exploiting the proposed DPA to capture or crash the drone (Section V-G). We leave it as future work to improve the transferability and understand the commercial ATT systems to further investigate the real-world security problems.

VII. Conclusion

In this paper, we conduct the first systematic study on the security of camera-based Autonomous Target Tracking (ATT) systems, with a focus on the newly proposed physical-world distance-pulling attacks (DPA). We define the problem with domain-specific goals and introduce the adversarial umbrella as a novel, real-world deployable attack vector. By designing a progressive distance-pulling strategy and controllable spatial-temporal consistency, we achieve closed-loop distance-pulling effects and spatial-temporal consistent attacks. Through a new dataset and system-level metrics, we demonstrate the attack's high effectiveness and generalizability. Physical-world evaluations with adversarial umbrella prototypes and full-stack ATT drones, alongside black-box testing on commercial drones, reveal significant real-world implications. Given the critical security and safety concerns surrounding ATT, we hope our findings will inspire future research and community attention.

VIII. Ethics Considerations

Disclosures. We evaluated our attack on three commercial drones and confirmed that their ATT features are all vulnerable to the FlyTrap attack, sometimes causing high-speed collisions. To prevent negative impacts, we have performed responsible vulnerability disclosure to both affected manufacturers prior to all public releases of this work, following the ethical standards in the security community.

Data collection. We collected videos of researchers (with obscured facial features) to evaluate ML models for tracking and detection. No identifiable private information was involved, and participants could not be identified. As confirmed by our IRB officers, this anonymized setup does not qualify as human subject research under federal policy [88].

User study. We conducted anonymous visual recognition tasks on Prolific. According to the Federal Policy [86], this qualifies for exempt Category-2 [87] since no identifiable information was collected and participants faced no physical or psychological risks. Thus, IRB review was not required.

Acknowledgment

We would like to thank Chi Zhang, Li-Chen Cheng, and the anonymous reviewers for their valuable and insightful feedback. This work was supported by the National Science Foundation under grant CNS-2145493 and by the NASA University Leadership Initiative under Award 80NSSC24M0070. All drone data and experiments presented in this work were completed before December 22, 2025.

References

[1] Andrew Adams, "Illinois Expands Use of Police Surveillance Drones," https://capitolnewsillinois.com/news/illinois-expands-use-of-police-surveillance-drones, 2023.

[2] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in International Conference on Machine Learning (ICML), 2018.

[3] AUTEL, "What Is Autel EVO 2 Dynamic Track 2.0?" https://www.autelpilot.com/blogs/news/what-is-autel-evo-2-dynamic-track-2-0, 2021.

[4] AutelPilots Forum, "Dynamic Track Distance Limitations," https://autelpilots.com/threads/dynamic-track-distance.7954/, 2024.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-Convolutional Siamese Networks for Object Tracking," in European Conference on Computer Vision (ECCV), 2016.

[6] Billy Kyle, "DJI Mini 4 Pro ActiveTrack 360º Full Tutorial - A Brand New Experience," https://www.youtube.com/watch?v=xM_d0FBnweY, 2024.

[7] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control. John Wiley & Sons, 2015.

[8] 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 Symposium on Security and Privacy (SP). IEEE, 2021.

[9] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial Sensor Attack on LiDAR-Based Perception in Autonomous Driving," in ACM SIGSAC Conference on Computer and Communications Security (CCS), 2019.

[10] A. Chakrabarty, R. Morris, X. Bouyssounouse, and R. Hunt, "Autonomous Indoor Object Tracking with the Parrot AR.Drone," in International Conference on Unmanned Aircraft Systems (ICUAS), 2016.

[11] X. Chen, C. Fu, F. Zheng, Y. Zhao, H. Li, P. Luo, and G.-J. Qi, "A Unified Multi-Scenario Attacking Network for Visual Object Tracking," in Proceedings of the AAAI Conference on Artificial Intelligence, 2021.

[12] X. Chen, X. Yan, F. Zheng, Y. Jiang, S.-T. Xia, Y. Zhao, and R. Ji, "One-Shot Adversarial Attacks on Visual Tracking with Dual Attention," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[13] Y. Chen, Z. Li, L. Li, S. Ma, F. Zhang, and C. Fan, "An Anti-Drone Device based on Capture Technology," Biomimetic Intelligence and Robotics, 2022.

[14] H. Cheng, L. Lin, Z. Zheng, Y. Guan, and Z. Liu, "An Autonomous Vision-Based Target Tracking System for Rotorcraft Unmanned Aerial Vehicles," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017.

[15] H.-M. Chuang, D. He, and A. Namiki, "Autonomous Target Tracking of UAV Using High-Speed Visual Feedback," Applied Sciences, 2019.

[16] M. Contributors, "MMCV: OpenMMLab Computer Vision Foundation," https://github.com/open-mmlab/mmcv, 2018.

[17] Y. Cui, C. Jiang, L. Wang, and G. Wu, "MixFormer: End-to-End Tracking with Iterative Mixed Attention," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[18] David Hambling, "Forbes: Ukraine Rolls Out Target-Seeking Terminator Drones," https://www.forbes.com/sites/davidhambling/2024/03/21/ukraine-rolls-out-target-seeking-terminator-drones.

[19] Defense Central, "Capturing Enemy Drones Using Nets?" https://www.youtube.com/watch?v=fDftjuGM5mM, 2024.

[20] 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 Proceedings of the AAAI Conference on Artificial Intelligence, 2021.

[21] DJI, "DJI Mavic 3 Pro," https://store.dji.com/product/dji-mavic-3-pro?vid=137691, 2023.

[22] DJI, "DJI Mini 4 Pro," https://www.dji.com/mini-4-pro?site=brandsite&from=homepage, 2024.

[23] DJI, "DJI Mobile SDK: DJIActiveTrackTrackingState Class Reference," https://developer.dji.com/api-reference/android-api/Components/Missions/DJIActiveTrackTrackingState.html, 2025.

[24] DJI, "DJI Neo," https://www.dji.com/neo, 2025.

[25] DJI Forum, "DJI Active Tracking Distance Restrictions," https://forum.dji.com/forum.php?mod=redirect&goto=findpost&ptid=284247&pid=2972409, 2023.

[26] DroneXL, "Skydio 2 Software Update Offers Many New Features," https://dronexl.co/2020/06/25/skydio-2-software-update#:~:text=Maximum%20tracking%20distance%20with%20the%20phone%20has%20increased%20from%2010%20to%2020%20meters%20(66%20feet), 2020.

[27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, 2010.

[28] C. Express, "Clover: ROS-Based Framework and Raspberry Pi Image to Control PX4-Powered Drones," https://github.com/CopterExpress/clover, 2025.

[29] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[30] H. Fradi, L. Bracco, F. Canino, and J.-L. Dugelay, "Autonomous Person Detection and Tracking Framework Using Unmanned Aerial Vehicles (UAVs)," in European Signal Processing Conference (EUSIPCO). IEEE, 2018.

[31] C. Fu, S. Li, X. Yuan, J. Ye, Z. Cao, and F. Ding, "AD^2 Attack: Adaptive Adversarial Attack on Real-Time UAV Tracking," in International Conference on Robotics and Automation (ICRA). IEEE, 2022.

[32] M. García, R. Caballero, F. González, A. Viguria, and A. Ollero, "Autonomous Drone with Ability to Track and Capture an Aerial Target," in International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2020.

[33] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," International Conference on Learning Representations (ICLR), 2015.

[34] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering Adversarial Images Using Input Transformations," International Conference on Learning Representations (ICLR), 2018.

[35] X. Han, H. Wang, K. Zhao, G. Deng, Y. Xu, H. Liu, H. Qiu, and T. Zhang, "VisionGuard: Official Code Repository," https://zenodo.org/records/11140958, 2024.

[36] X. Han, H. Wang, K. Zhao, G. Deng, Y. Xu, H. Liu, H. Qiu, and T. Zhang, "VisionGuard: Secure and Robust Visual Perception of Autonomous Vehicles in Practice," in ACM SIGSAC Conference on Computer and Communications Security, 2024.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[38] S. Hochreiter, "Long Short-Term Memory," Neural Computation MIT-Press, 1997.

[39] Holybro, "PX4 Development Kit X500 V2," https://holybro.com/collections/x500-kits/products/px4-development-kit-x500-v2, 2025.

[40] HOVERAir, "HOVERAir Camera Drone Ultra-Light Self-Piloting," https://www.amazon.com/HOVERAir-Camera-Drone-Ultra-Light-Self-Piloting/dp/B0DLGRP5PQ, 2025.

[41] A. G. Howard, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.

[42] S. Ingram, "Canonsburg Man Charged with Stalking After Allegedly Using Drone to Follow Woman," https://www.wtae.com/article/canonsburg-drone-stalking-washington-county/61100793, 2024.

[43] Intel Corporation, "Intel RealSense Depth Camera D435i," https://www.intelrealsense.com/depth-camera-d435i/, 2025.

[44] X. Ji, Y. Cheng, Y. Zhang, K. Wang, C. Yan, W. Xu, and K. Fu, "Poltergeist: Acoustic Adversarial Machine Learning Against Cameras and Computer Vision," in 2021 IEEE Symposium on Security and Privacy (SP). IEEE, 2021.

[45] S. Jia, C. Ma, Y. Song, and X. Yang, "Robust Tracking against Adversarial Attacks," in European Conference on Computer Vision (ECCV). Springer, 2020.

[46] S. Jia, Y. Song, C. Ma, and X. Yang, "IoU Attack: Towards Temporally Coherent Black-Box Adversarial Attack for Visual Object Tracking," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[47] 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," Network and Distributed System Security (NDSS) Symposium, 2022.

[48] R. Jiao, H. Liang, T. Sato, J. Shen, Q. A. Chen, and Q. Zhu, "End-to-End Uncertainty-Based Mitigation of Adversarial Attacks to Automated Lane Centering," in IEEE Intelligent Vehicles Symposium (IV). IEEE, 2021.

[49] John, Davis, "U.S. Customs and Border Protection: Small but Mighty: Border Patrol's use of small drones is a game changer in border security," https://www.cbp.gov/frontline/cbp-small-drones-program.

[50] Joseph Trevithick, "New Jersey Base Confirms Multiple Past Drone Incursions… By Contraband Smugglers," https://news.yahoo.com/news/jersey-confirms-multiple-past-drone-211029636.html, 2024.

[51] A. G. Kendall, N. N. Salvapantula, and K. A. Stol, "On-Board Object Tracking Control of a Quadcopter with Monocular Vision," in International Conference on Unmanned Aircraft Systems (ICUAS), 2014.

[52] Khari Johnson, "This New Autonomous Drone for Cops Can Track You in the Dark," https://www.wired.com/story/new-autonomous-drone-for-cops-can-track-you-in-the-dark/, 2023.

[53] N. Koenig and A. Howard, "Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2004.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems (NeurIPS), 2012.

[55] A. Levine and S. Feizi, "(De)Randomized Smoothing for Certifiable Defense Against Patch Attacks," Advances in Neural Information Processing Systems (NeurIPS), 2020.

[56] 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 (CVPR), 2019.

[57] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High Performance Visual Tracking with Siamese Region Proposal Network," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[58] S. Liang, X. Wei, S. Yao, and X. Cao, "Efficient Adversarial Attacks for Visual Object Tracking," in European Conference on Computer Vision (ECCV). Springer, 2020.

[59] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in European Conference on Computer Vision (ECCV). Springer, 2014.

[60] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," in European Conference on Computer Vision (ECCV). Springer, 2016.

[61] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into Transferable Adversarial Examples and Black-Box Attacks," International Conference on Learning Representations (ICLR), 2016.

[62] Z. Liu, Y. Yuan, S. Wang, X. Xie, and L. Ma, "Decompiling x86 Deep Neural Network Executables," in 32nd USENIX Security Symposium (USENIX Security), 2023.

[63] W. Ma, Y. Li, X. Jia, and W. Xu, "Transferable Adversarial Attack for Both Vision Transformers and Convolutional Networks via Momentum Integrated Gradients," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023.

[64] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," arXiv Preprint arXiv:1706.06083, 2017.

[65] Y. Man, R. Muller, M. Li, Z. B. Celik, and R. Gerdes, "PercepGuard: Official Code Repository," https://github.com/Harry1993/PercepGuard, 2023.

[66] 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 32nd USENIX Security Symposium (USENIX Security), 2023.

[67] Mariano, Zafra and Max, Hunder and Anurag, Rao and Sudev, Kiyada, "Reuters: How Drone Combat in Ukraine is Changing Warfare," https://www.reuters.com/graphics/UKRAINE-CRISIS/DRONES/dwpkeyjwkpm/.

[68] Marvelous Catawba Otter, "A Brief Discussion: The Computational Cost of Backward Propagation Is Approximately Twice That of Forward Propagation," https://medium.com/@marvelous_catawba_otter_200/a-brief-discussion-the-computational-cost-of-backward-propagation-is-approximately-twice-that-of-5dd0eac9b389, 2023.

[69] R. Muller, Y. Man, Z. B. Celik, M. Li, and R. Gerdes, "Physical Hijacking Attacks Against Object Trackers," in ACM SIGSAC Conference on Computer and Communications Security (CCS), 2022.

[70] R. Muller, Y. Man, M. Li, R. Gerdes, J. Petit, and Z. B. Celik, "VOGUES: Official Code Repository," https://github.com/purseclab/VOGUES, 2024.

[71] R. Muller, Y. Man, M. Li, R. Gerdes, J. Petit, and Z. B. Celik, "VOGUES: Validation of Object Guise using Estimated Components," in 33rd USENIX Security Symposium (USENIX Security), 2024.

[72] K. K. Nakka and M. Salzmann, "Universal, Transferable Adversarial Perturbations for Visual Object Trackers," in European Conference on Computer Vision (ECCV). Springer, 2022.

[73] NetGun, "NetGun Official Website," https://www.net-gun.com/.

[74] NetGun, "UltraNet HD – Large Animal Target Net Gun," https://netgun.com/netgun-info/ultranet-hd-large-animal-target-net-gun.

[75] NVIDIA, "NVIDIA Jetson Nano," https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development, 2019.

[76] Paraben Corporation, "Drone Forensics: Navigating the New Frontier of Digital Evidence," https://paraben.com/drone-forensics-navigating-the-new-frontier-of-digital-evidence/?utm_source=chatgpt.com, 2024.

[77] People Staff, "Two Arrested for Flying Drones Dangerously Close to Logan Airport, Boston Police Report," https://people.com/drones-two-arrested-dangerously-close-logan-airport-boston-police-8761977, 2024.

[78] J. Pestana, J. L. Sanchez-Lopez, P. Campoy, and S. Saripalli, "Vision-based GPS-denied object tracking and following for unmanned aerial vehicles," in IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2013.

[79] J. Pestana, J. L. Sanchez-Lopez, S. Saripalli, and P. Campoy, "Computer Vision-Based General Object Following for GPS-Denied Multi-

rotor Unmanned Vehicles," in American Control Conference. IEEE, 2014.

[80] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR," Black Hat Europe, 2015.

[81] Potensic, "How to Better Use Potensic's Atom's Visual Tracking," https://store.potensic.com/blogs/news/how-to-better-use-atoms-visual-tracking?srsltid=AfmBOoqWRqzrmDs16OWVJwVc6qtEd35r3qeXj82le2gPjJCAuXZ8eBrl.

[82] Prolific, "Prolific: Participant Recruitment for Research," https://www.prolific.com, 2025.

[83] PX4 Contributors, "PX4 Autopilot," https://github.com/PX4/PX4-Autopilot.

[84] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng et al., "ROS: An Open-Source Robot Operating System," in ICRA Workshop on Open Source Software. Kobe, Japan, 2009.

[85] J. Redmon, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[86] F. Register, "Federal Register Document 2017-01058, Paragraph 1315," https://www.federalregister.gov/d/2017-01058/p-1315, 2017.

[87] F. Register, "Federal Register Document 2017-01058, Paragraph 1375," https://www.federalregister.gov/d/2017-01058/p-1375, 2017.

[88] F. Register, "Federal Register Document 2017-01058, Paragraph 203," https://www.federalregister.gov/d/2017-01058/p-203, 2017.

[89] Reuters, "Explainer: How Drones Are Being Used in the Ukraine War," https://www.reuters.com/graphics/UKRAINE-CRISIS/DRONES/dwpkeyjwkpm/, 2024.

[90] A. Robotics, "Autel EVO 2," https://shop.autelrobotics.com/collections/autel-evo-ii-series, 2020.

[91] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[92] T. Sato, Y. Hayakawa, R. Suzuki, Y. Shiiki, K. Yoshioka, and Q. A. Chen, "LiDAR Spoofing Meets the New-Gen: Capability Improvements, Broken Assumptions, and New Attack Strategies," Network and Distributed System Security (NDSS) Symposium, 2024.

[93] 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 30th USENIX Security Symposium (USENIX Security), 2021.

[94] T. Sato, J. Yue, N. Chen, N. Wang, and Q. A. Chen, "Intriguing Properties of Diffusion Models: An Empirical Study of the Natural Attack Capability in Text-to-Image Generative Models," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), 2024.

[95] N. Schiller, M. Chlosta, M. Schloegel, N. Bars, T. Eisenhofer, T. Scharnowski, F. Domke, L. Schönherr, and T. Holz, "Drone Security and the Mysterious Case of DJI's DroneID," in Network and Distributed System Security (NDSS) Symposium, 2023.

[96] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial Training for Free!" Advances in Neural Information Processing Systems (NeurIPS), 2019.

[97] Skydio, "Skydio 2 Plus," https://www.skydio.com/skydio-2-plus-enterprise, 2022.

[98] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Choi, and Y. Kim, "Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors," in 24th USENIX Security Symposium (USENIX Security), 2015.

[99] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild," arXiv preprint arXiv:1212.0402, 2012.

[100] T. N. G. Store, "The Net Gun Mega Pack – Most Popular," https://thenetgunstore.com/products/the-net-gun-mega-pack-most-popular?variant=9806433189945, 2025.

[101] Thomas, Brewster, "Forbes: Founded By Ex-Google Engineers, Meet The Drone Startup Scoring Millions In Government Surveillance Contracts," https://www.forbes.com/sites/thomasbrewster/2020/06/03/funded-by-kevin-durant-and-founded-by-ex-googlers-this-drone-startup-is-scoring-millions-in-government-surveillance-contracts/.

[102] Y. Tu, Z. Lin, I. Lee, and X. Hei, "Injected and Delivered: Fabricating Implicit Control Over Actuation Systems by Spoofing Inertial Sensors," in 27th USENIX Security Symposium (USENIX Security), 2018.

[103] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," Advances in Neural Information Processing Systems (NeurIPS), 2017.

[104] Z. Wan, J. Shen, J. Chuang, X. Xia, J. Garcia, J. Ma, and Q. A. Chen, "Too Afraid to Drive: Systematic Discovery of Semantic DoS Vulnerability in Autonomous Driving Planning Under Physical-World Attacks," in Network and Distributed System Security (NDSS) Symposium, 2022.

[105] N. Wang, Y. Luo, T. Sato, K. Xu, and Q. A. Chen, "Does Physical Adversarial Example Really Matter to Autonomous Driving? Towards System-Level Effect of Adversarial Object Evasion Attack," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023.

[106] N. Wang, S. Xie, T. Sato, Y. Luo, K. Xu, and Q. A. Chen, "Revisiting Physical-World Adversarial Attack on Traffic Sign Recognition: A Commercial Systems Perspective," in Network and Distributed System Security (NDSS) Symposium, 2025.

[107] 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 (ICCV), 2019.

[108] R. Wu, T. Kim, D. J. Tian, A. Bianchi, and D. Xu, "DnD: A cross-architecture deep neural network decompiler," in 31st USENIX Security Symposium (USENIX Security), 2022.

[109] C. Xiang, A. N. Bhagoji, V. Sehwag, and P. Mittal, "PatchGuard: A Provably Robust Defense Against Adversarial Patches via Small Receptive Fields and Masking," in 30th USENIX Security Symposium (USENIX Security), 2021.

[110] C. Xiang and P. Mittal, "DetectorGuard: Provably Securing Object Detectors Against Localized Patch Hiding Attacks," in ACM SIGSAC Conference on Computer and Communications Security (CCS), 2021.

[111] C. Xiang and P. Mittal, "PatchGuard++: Efficient Provable Attack Detection Against Adversarial Patches," in ICLR Workshop on Security and Safety in Machine Learning Systems, 2021.

[112] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating Adversarial Effects Through Randomization," International Conference on Learning Representations (ICLR), 2018.

[113] S. Xie, M. H. Fakih, J. Lu, F. Alshammari, N. Wang, T. Sato, H. Bouzidi, M. A. Al Faruque, and Q. A. Chen, "Flytrap Project Website," https://sites.google.com/view/av-ioat-sec/flytrap, 2025.

[114] Y. Xu, G. Deng, X. Han, G. Li, H. Qiu, and T. Zhang, "PhyScout: Detecting Sensor Spoofing Attacks via Spatio-Temporal Consistency," in ACM SIGSAC Conference on Computer and Communications Security (CCS), 2024.

[115] B. Yan, D. Wang, H. Lu, and X. Yang, "Cooling-Shrinking Attack: Blinding the Tracker with Imperceptible Noises," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[116] C. Yan, W. Xu, and J. Liu, "Can You Trust Autonomous Vehicles: Contactless Attacks Against Sensors of Self-Driving Vehicle," Def Con, 2016.

[117] X. Yan, X. Chen, Y. Jiang, S.-T. Xia, Y. Zhao, and F. Zheng, "Hijacking Tracker: A Powerful Adversarial Attack on Visual Tracking," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

[118] X. Yin, W. Ruan, and J. Fieldsend, "DIMBA: Discretely Masked Black-Box Attack in Single Object Tracking," Machine Learning, 2024.

[119] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[120] Z. Yu, A. Li, R. Wen, Y. Chen, and N. Zhang, "Physense: Defending Physically Realizable Attacks for Autonomous Systems via Consistency Reasoning," in ACM SIGSAC Conference on Computer and Communications Security (CCS), 2024.

[121] H. Zhang, G. Wang, Z. Lei, and J.-N. Hwang, "Eye in the Sky: Drone-Based Object Tracking and 3D Localization," in Proceedings of the 27th ACM International Conference on Multimedia (MM), 2019.

[122] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On Adversarial Robustness of Trajectory Prediction for Autonomous Vehicles," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[123] 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 SIGSAC Conference on Computer and Communications Security (CCS), 2019.
[124] C. Zhou, Q. Yan, Y. Shi, and L. Sun, "DoubleStar: Long-Range Attack Towards Depth Estimation-Based Obstacle Avoidance in Autonomous Systems," in 31st USENIX Security Symposium (USENIX Security), 2022.

APPENDIX

A. Human Figure Printing Baseline Attack

We introduce the target photo baseline attack (TGT), where attackers print their own photos on an umbrella to misguide the ATT drone (Fig. 16). TGT is straightforward: (1) printed images resemble the tracked target, (2) their smaller size induces a shrinking effect, and (3) it requires no adversarial ML expertise. To ensure fair comparison with FlyTrap, we grid-search the person/background ratio maximizing $\text{mASR}_{\text{open}}$. If too large, the shrinkage is weak; if too small, the features are insufficient. Using four people and four backgrounds, we generate TGTs with ratios 0.01–0.9, then test across four models to pick the optimal ratio.

Fig. 16: Target photo baseline attack (TGT). Attackers print a self-photo to mislead the tracker. The printed figure resembles the target and appears smaller than the original. Distortion is from rendering to simulate umbrella geometry.

B. Real-World Attack Distance and Angle Discussion

The maximum attack distance of FlyTrap is related to the ATT tracking limit, determined mainly by the drone's camera resolution and optics. Consumer drones (e.g., DJI Mini, Potensic, Autel, Skydio 2) with 4K video typically track humans up to ∼20m [25], [81], [4], [26]. In our experiments with DJI Mini 4 Pro, FlyTrap remains fully effective up to 20m and, in some cases, generalizes to 30–40m, beyond its training range. Thus, within the tested scope, FlyTrap shows no attack distance limitation, though longer-range validation is left for future work.

For the attack angle, we simulate umbrella misalignment by varying azimuth/elevation during rendering. FlyTrap tolerates pointing errors within $\pm 10°$ (Fig. 17), with significant degradation only beyond 30°, which we consider to be already well within the normal controllable range when a normal person intentionally tries to aim the umbrella at the drone (visualized in Fig. 18). Real-world tests in Section V-F also confirmed that best-effort aiming is sufficient, requiring no retries to achieve the reported attack effects.

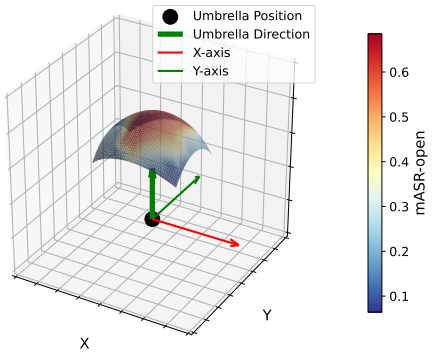


Fig. 17: Spatial view of $\text{mASR}_{\text{open}}$ under angle variation.

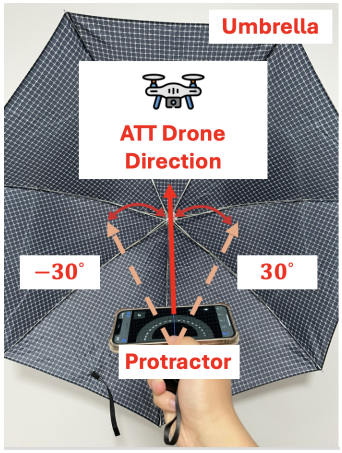


Fig. 18: A real-world example of umbrella aiming. As shown, $\pm 30°$ is within the normal controllable range when a normal person intentionally tries to aim the umbrella at the drone.

C. Full-Stack ATT System Implementation

Hardware setup. Our platform is the Holybro X500 v2 quadcopter [39] with a Pixhawk flight controller (inertial sensors + GPS). An NVIDIA Jetson Orin Nano [75] handles detection, tracking, and control, while PX4 [83] runs on the controller. An Intel RealSense D435i camera [43] provides video, streamed via a Jetson-hosted WiFi interface where operators select a target by drawing a bounding box (Fig. 10).
Software setup. We use ROS [84] with four nodes: `UserUINode`, `DetectorNode`, `TrackerNode`, and `ControlNode`. The user selects a target through the web UI; `DetectorNode` refines the bounding box using SSD-MobileNet-V2 [60], [91], filtering for the "person" class and choosing the highest-IoU box. This improves SOT accuracy, especially when manual selection is imprecise. `TrackerNode` then tracks the target, outputting bbox and confidence. `ControlNode` acts only if confidence exceeds a threshold, computing movement offsets $(\Delta x, \Delta y, \Delta z)$ to keep the target centered and sized consistently. Then, we apply offsets to the current pose (x, y, z) in real time for stable motion.

ARTIFACT APPENDIX

D. Description

FLYTRAP is a physical distance-pulling attack that dangerously reduces the effective range of autonomous target tracking (ATT) systems, such as those used in visual tracking drones. By executing a Distance-Pulling Attack (DPA), an adversary can conduct range-limited sensor attacks, capturing, or even direct crashes of autonomous drones.

This artifact contains the full codebase, pre-trained models, optimized adversarial patches, and the dataset used to implement and evaluate the proposed FLYTRAP attack. We provide detailed instructions for environment setup, execution of evaluation pipelines, and reproduction of all experimental results presented in the paper.

E. Requirements

Access: The artifact is publicly available on GitHub at https://github.com/Daniel-xsy/FlyTrap. The repository includes a comprehensive `README.md` file that provides step-by-step setup instructions, experiment configurations, and command-line examples for reproducing our results. We also upload the following materials onto the Zenodo platform:

- Codebase: https://doi.org/10.5281/zenodo.17051835
- Dataset: https://doi.org/10.5281/zenodo.16908024
- Models: https://doi.org/10.5281/zenodo.17051654

Hardware Requirements: A GPU is required to run the experiments. We recommend a minimum of 24 GB of GPU memory. All evaluations were conducted on a machine with two NVIDIA RTX 3090 GPUs, though the artifact can be run on a single GPU with increased execution time. The system CPU used was an AMD EPYC 7513 32-Core Processor.

Software Requirements: The artifact mainly uses `PyTorch` deep learning framework. All experiments were executed on Ubuntu 20.04, with `PyTorch=1.11` and `CUDA=11.3`.

Storage Requirements: The artifact requires approximately 20 GB of disk space. This includes around 16 GB for the dataset and 2 GB for the pre-trained victim Single-Object Tracking (SOT), object detection, and pose estimation model checkpoints. The remainder is allocated for other materials.

F. Codebase Design

The codebase is designed to be modular, extensible, and scalable. It follows a registry-based architecture, enabling flexible training and evaluation workflows. All experiments can be launched using a single configuration file. The major components of the codebase are organized as follows:

`./config`: Contains configuration files used to optimize and evaluate the FLYTRAP attack. Each file specifies victim model hyperparameters, the data loading pipeline, loss objectives, and the physical simulation engine, which together compose a complete pipeline for adversarial patch generation.

`./flytrap`: Implements core functionality. We leverage the `mmcv` [16] registry system to modularize components, allowing easy integration via configuration files. This includes:

- `attacks`: Defines the attack pipeline, including modules for digital rendering and patch application, as well as loss functions used for optimization.
- `dataset`: Implements the data loading pipeline required for adversarial patch optimization.
- `engine`: Simulates the closed-loop drone tracking behavior, supporting our Progressive Distance-Pulling design (PDP) and attack target control.
- `metrics`: Defines $\text{mASR}_{\text{open}}$ metrics in evaluate the.
- `models`: Provides unified API wrappers to build victim models from their original implementations.

`./models`: Contains the original implementations of third-party tracking models used as victims in our evaluation.

`./tools`: Includes entry-point scripts for executing optimization and evaluation procedures.

G. Major Claims

The provided artifact supports the validation of the key experimental claims presented in the paper. All necessary code, configurations, and resources are included to facilitate reliable replication of these core findings.

[**C1: Table II**]: FLYTRAP with Progressive Distance-Pulling (PDP) achieve higher effectiveness than FLYTRAP w/o PDP.

[**C2: Table II and III**]: FLYTRAP can achieve better effectiveness and universality than target image baseline attack (TGT).

[**C3: Table IV and V**]: FLYTRAP with attack target generation (ATG) design can decrease the true alarm rate (TAR) of spatial-temporal consistency defenses.

[**C4: Table VI**]: FLYTRAP with ATG design does not largely reduce the attack effectiveness.

For the physical experiments, we provide recorded demonstration videos along with corresponding evaluation scripts. Due to hardware dependencies, such as the need for our implemented drone platform, the commercial drone platform, and a physical adversarial umbrella, we do not include closed-loop physical experiments in this artifact.

H. Evaluation

1) Experiment (E1): [**C1**] [5 human-minutes + 4 computer hours]: This experiment tests the Progressive Distance-Pulling (PDP) design of the FlyTrap attack.

- *Full Evaluation*: [5 human-minutes + 4 computer hours]. Please run the command:
  ```
  bash scripts/eval_flytrap.sh
  bash scripts/metric_summary.sh
  ```
- *Partial Evaluation*: [5 human-minutes + 1 computer hours]. Please run the command:
  ```
  python tools/main.py <config>
  cd analysis
  python analyze_result_metric.py --file
  <result_path>
  ```
- *Pre-computed Evaluation*: [5 human-minutes + 5 computer minutes]. Please run the command:
  ```
  cd download
  bash download_flytrap_results.sh
  cd ../
  ```

```
bash scripts/metric_summary.sh
```

2) Experiment (E2): [**C2**] [5 human-minutes + 40 computer hours]: This experiment compares the FlyTrap attack with baseline target photo attack.

- *Full Evaluation*: [5 human-minutes + 40 computer hours]. Please run the command:
  `bash scripts/eval_tgt.sh`
- *Partial Evaluation*: [5 human-minutes + 1 computer hours]. Please run the command:
  `bash scripts/eval_tgt_partial.sh <config>`
- *Pre-computed Evaluation*: [5 human-minutes + 5 computer minutes]. Please run the command:
  `bash download/download_tgt_results.sh`

Then, run the command for evaluation:
`python analysis/analyze_tgt_metric.py --input_dir <json_dir>`

3) Experiment (E3): [**C3**] [5 human-minutes + 4 computer hours]: This experiment compares the true alarm rate of PercepGuard defense before and after applying attack target generation (ATG).

- *Full Evaluation*: [5 human-minutes + 4 computer hours]. Please run the command, the `config` and `adv_patch` please refer to the GitHub repository:
  `bash scripts/eval_percepguard.sh <config> <adv_patch>`
- *Partial Evaluation*: [5 human-minutes + 1 computer hour]. You can only compare the results of one model:
  `bash scripts/eval_percepguard.sh <config> <adv_patch>`

Please average across all the model results to reproduce the results in the paper.

4) Experiment (E4): [**C3**] [5 human-minutes + 10 computer hours]: This experiment compares the true alarm rate of VOGUES defense before and after applying attack target generation (ATG).

- *Full Evaluation*: [5 human-minutes + 10 computer hours]. Please run the command, the `config` and `adv_patch` please refer to the GitHub repository:
  `bash scripts/eval_vogues.sh <config> <adv_patch>`
- *Partial Evaluation*: [5 human-minutes + 2.5 computer hour]. You can only compare the results of one model:
  `bash scripts/eval_vogues.sh <config> <adv_patch>`

The JSON file results will be saved in this directory: `work_dirs/vogues_results`. For the results:

- With Attack: `before` means the false alarm rate before the attack, and `after` means the true alarm rate after the attack.
- Without attack: `before` means the false alarm rate without the umbrella (should be the same as above), and `after` means the false alarm rate with the umbrella.

5) Experiment (E5): [**C4**] [10 human-minutes + 10 computer hours]: This experiment compares the FlyTrap attack with and without ATG design. Please refer to FlyTrap without ATG design in *E1*: $\text{FlyTrap}_{\text{PDP}}$. Please evaluate $\text{FlyTrap}_{\text{ATG}}$ results by running:

```
bash scripts/eval_flytrap_atg.sh
bash scripts/metric_summary_atg.sh
```