Physical ID-Transfer Attacks against Multi-Object Tracking via Adversarial Trajectory

Chenyi Wang
University of Arizona
chenyiw@arizona.edu

Yanmao Man
HERE Technologies
yman@arizona.edu

Raymond Muller
Purdue University
mullerr@purdue.edu

Ming Li
University of Arizona
lim@arizona.edu

Z. Berkay Celik
Purdue University
zcelik@purdue.edu

Ryan Gerdes
Virginia Tech
rgerdes@vt.edu

Jonathan Petit
Qualcomm
petit@qti.qualcomm.com

***Abstract*—Multi-Object Tracking (MOT) is a critical task in computer vision, with applications ranging from surveillance systems to autonomous driving. However, threats to MOT algorithms have yet been widely studied. In particular, incorrect association between the tracked objects and their assigned IDs can lead to severe consequences, such as wrong trajectory predictions. Previous attacks against MOT either focused on hijacking the trackers of individual objects, or manipulating the tracker IDs in MOT by attacking the integrated object detection (OD) module in the digital domain, which are model-specific, non-robust, and only able to affect specific samples in offline datasets. In this paper, we present ADVTRAJ, the first online and physical ID-manipulation attack against tracking-by-detection MOT, in which an attacker uses adversarial trajectories to transfer its ID to a targeted object to confuse the tracking system, without attacking OD. Our simulation results in CARLA show that ADVTRAJ can fool ID assignments with 100% success rate in various scenarios for white-box attacks against SORT, which also have high attack transferability (up to 93% attack success rate) against state-of-the-art (SOTA) MOT algorithms due to their common design principles. We characterize the patterns of trajectories generated by ADVTRAJ and propose two universal adversarial maneuvers that can be performed by a human walker/driver in daily scenarios. Our work reveals under-explored weaknesses in the object association phase of SOTA MOT systems, and provides insights into enhancing the robustness of such systems.**

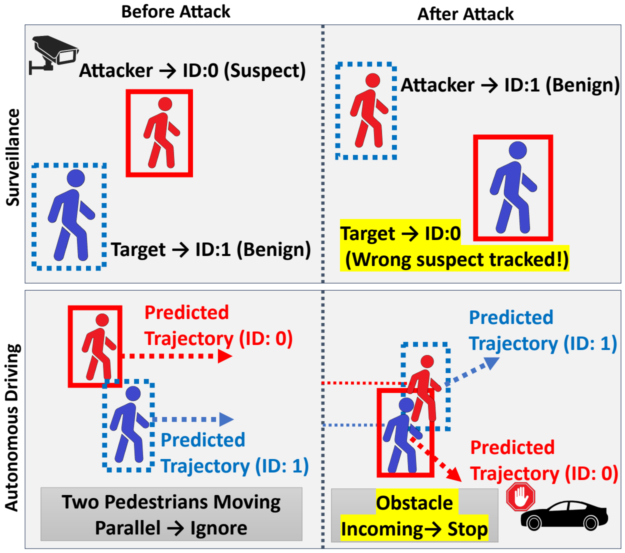


Figure 1: Illustration of potential consequences of ID-Transfer in surveillance and autonomous driving (AD) applications. In the surveillance scenario (above), ID-Transfer leads to wrong target of interest being tracked. In the AD scenario (below), ID-Transfer results in inaccurate trajectory prediction due to history trajectories that are inconsistent with ground truths.

1. Introduction

In computer vision, Multi-Object Tracking (MOT) algorithms play a pivotal role in understanding and interpreting dynamic scenes. These algorithms are designed to track multiple objects simultaneously by assigning unique IDs as they move across video frames. With applications ranging from autonomous driving (AD) [1–4] and pedestrian/vehicle surveillance systems [5] to military unmanned aerial vehicles [6, 7], the assigned IDs of the MOT system are used to uniquely identify objects of interest for trajectory predictions in AD and suspect tracking in surveillance systems. Due to the safety-critical nature of these applications, the correct and consistent association between assigned IDs and tracked objects is of crucial importance. For example, in surveillance systems, as shown in Figure 1, accurate and consistent identification allows effective monitoring and timely response to incidents. An ID mismatch can lead to losing track of the object of interest, resulting in wrongful accusations or the escape of tracking by a criminal. Furthermore, AD systems (e.g., Baidu Apollo [1]) typically operate through a pipeline that includes perception, object tracking, trajectory prediction, planning, and control. If tracked objects' IDs are mismatched, the prediction module will make wrong trajectory predictions based on incorrect trajectory histories.

Most state-of-the-art (SOTA) MOT algorithms (e.g., ByteTrack [8], OC-SORT [9], etc.) generally follow the tracking-by-detection paradigm. This consists of two explicit stages: (1) an object detection (OD) phase that produces bounding box detections, and (2) an association phase that performs weighted bipartite matching between detections and bounding boxes produced by motion prediction models. The motion prediction model, such as Kalman Filter [10], maintained by individually indexed trackers, is an integrated and necessary component of MOT. This is because other information that can potentially be used for object identification (such as facial recognition or other biometric features) is not always available due to privacy concerns, different camera angles, low resolution, or different application scenarios (e.g., vehicle tracking), etc.

Although previous works have shown that MOT is vulnerable to several types of attacks, they have mainly targeted the OD component, and primarily aimed at inducing errors in an individual object's detected trajectory. For example, recent tracker hijacking attacks [11, 12] cause the detected trajectory to deviate from the true track of an object. Other works investigated identity-switching attacks that cause a target object's ID to be switched to a different one [13, 14], causing a loss of tracking on the target of interest. However, these works are highly dependent on traditional perturbation-based attacks against OD that require strong attacker capability, such as full read and write access to the video feed in an offline dataset and white-box knowledge of the OD model structure and parameters, which limits their practical impact.

In this paper, we introduce AdvTraj, a novel physical and online ID-Transfer attack that confuses the tracking of *two* objects, rather than introducing a tracking error on an individual object. Instead of attacking the OD models of MOT, our attack exploits the vulnerability of the association phase (especially motion prediction), by using physically realizable yet adversarial trajectories to confuse the ID assignment and matching algorithm. Unlike previous works, we consider a stealthier ID-manipulation attack in which the attacker aims to *transfer* its MOT-assigned ID to another targeted and non-cooperative object without losing the attacker's original track ID, because otherwise it could raise suspicion. In addition, we consider a more realistic threat model where the attack is online, physically realizable, and does not require digital modification of the video input.

We start by assuming an adversary with white-box knowledge of the MOT algorithm. By deriving conditions on desired ID assignments and optimizing for physical trajectories, AdvTraj addresses several technical challenges: (a) non-differentiability in the bipartite matching algorithm, (b) manipulating detected bounding boxes, and (c) adherence to physical constraints. Hence, the attack can be conducted in real-time where the perceived image sequence of the MOT system is genuine, but contains the adversarial trajectory. We implement our white-box attack against the SORT algorithm [15], and then show it can be readily transferable to other SOTA MOT algorithms under the black-box setting, due to their common design principles.

By investigating the patterns of adversarial trajectories generated by AdvTraj, we identify several underlying characteristics of these trajectories. Based on this, we further develop universal adversarial maneuvers (UAMs) that can be easily realized by a human walker/driver, since executing optimized adversarial trajectory in the real world would require precise motor control. These UAMs effectively introduce discrepancies between the attacker's actual and predicted locations when being close to the target, by performing a nonlinear movement (e.g., acceleration and deceleration), which induces ID-Transfer. Our contributions are summarized as follows:

- We introduce AdvTraj, the *first* online and physical ID-Transfer attack against MOT algorithms using an adversarial trajectory. AdvTraj enables an attacker to disrupt the MOT's ID assignment by transferring its own ID to another target, manipulating the system's ability to correctly track objects.
- We show that the white-box attack against SORT is also transferable to other SOTA MOT algorithms, due to their common design principles, which eliminates the requirement of adversary's knowledge of the MOT algorithm.
- We evaluate AdvTraj in CARLA for surveillance and autonomous driving applications. Our simulation evaluation demonstrates an attack success rate of 100% for white-box attacks against SORT, and up to 93% attack success rate for transferred black-box attacks against 5 other SOTA MOT algorithms.
- We characterize the patterns of optimized adversarial trajectories discovered using AdvTraj and propose two universal (heuristic) adversarial maneuvers that are easily realizable, which achieve up to 45% attack success rates in our real-world experiments.
- We discuss a set of potential countermeasures to mitigate the vulnerabilities in the association stage of MOT algorithms against ID-Transfer attacks.

2. Background and Related Work

2.1. Multi-Object Tracking

Object tracking (OT) is a fundamental task for analyzing image sequences. Compared to object detection (OD) algorithms, which perform object classification and localization on a static image, OT extends the task to consistently identify detected objects across frames by assigning unique IDs to distinct object instances. There are two categories for OT algorithms based on their goals. Single-object tracking (SOT) [16, 17] tracks a single object specified in the reference frame and provides localization results in subsequent frames. In contrast, multi-object tracking (MOT) [15, 18] simultaneously matches multiple detected objects with previous trajectories, offering extensive utilities with applications ranging from surveillance [5] and autonomous driving systems [1, 2], to unmanned aerial vehicles [6].

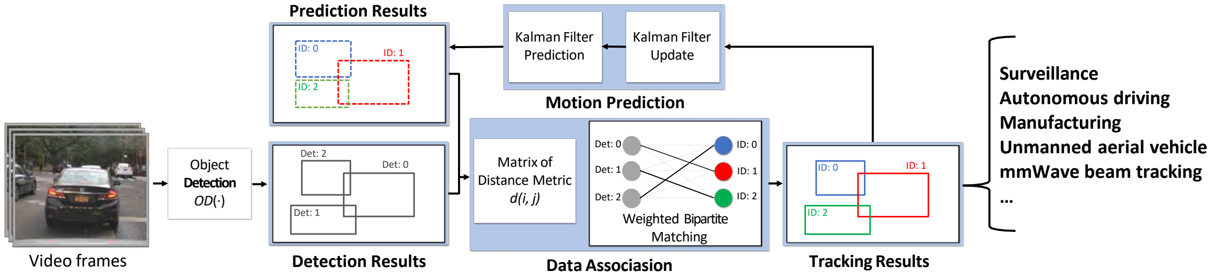


Figure 2: Illustration of tracking-by-detection pipeline in Multi-Object Tracking (MOT).

MOT algorithms can themselves be categorized into two paradigms: joint-detection-and-tracking and tracking-by-detection. The former is an end-to-end approach that aims to unify the detection and tracking processes into a single cohesive model [19–21]. On the other hand, the tracking-by-detection framework consists of two stages where the system identifies objects in each frame explicitly using OD models, and then makes associations across frames to form trajectories [9, 18, 22], which are more commonly used in autonomous systems [1, 4, 12].

Figure 2 shows the general pipeline of tracking-by-detection. Upon arrival of each frame, the system calls the OD model to locate detected objects and passes the bounding boxes to the tracking (i.e., association) module. For each tracked object, the tracker maintains a motion prediction model and optionally the Re-Identification (ReID) features, which use deep learning models to represent the object's appearance in past frames [23–25]. The motion prediction model in each tracker, usually implemented using Kalman Filter [10], is used to produce a predicted bounding box position based on the object's past state estimates. Finally, the tracking module performs weighted bipartite graph matching based on common distance metrics: (1) Intersection-over-Union (IoU) between detected and tracker-predicted bounding boxes, and/or (2) distance in ReID feature space.

2.2. Existing Attacks

Since the introduction of adversarial examples against image classification [26], various attacks have been proposed against object detection (OD) [27–31] and single-object tracking (SOT) [32–36]. Although OD is an integrated and vital stage of tracking-by-detection based MOT algorithms, it has been shown that an attack targeting OD (e.g., vanishing attack) needs to succeed at least 98% of the time over 60 consecutive frames to influence the MOT algorithms [11]. To date, no OD attack has been able to achieve such a high success rate. In addition, the real-world impact of attacks on SOT remains unclear, since most surveillance and AD systems adopt MOT. Although MOTs are used in safety-critical applications, there are only a limited number of MOT attacks due to the difficulty of directly applying OD attacks in the MOT pipeline.

Depending on the primary attack objectives, current MOT attacks can be broadly divided into two categories: (a) tracker hijacking and (b) identity switching. Tracker hijacking attacks [11, 12, 36] aim to mislead the tracking of a target object to an incorrect trajectory by suppressing the true detected bounding box while fabricating fake ones at adjacent but wrong locations. The attack can be applied in both SOT and MOT on a single target object. On the other hand, identity switching attacks [13, 14] are applied to MOT where the attacker manipulates the detected bounding boxes to switch the identity of a target object assigned by the MOT system to a different one. Although these two types of attacks differ in attack goals, they are evaluated using the same metric of ID-Switch, where an attack is considered successful if the target object switches to a new ID after the attack (while its old one is not necessarily preserved).

Jia et al. [11] proposed an offline tracker hijacking attack in which an attacker places an adversarial patch to make the detected object bounding box shift toward the direction opposite to its true movement, resulting in the original tracker being misled and eventually lost while the tracked object switches to a new identity after a certain number of frames. However, the attack was only demonstrated in the digital domain where only a single tracked object is present. The attack's practicality and robustness were questioned in their subsequent work in progress [12] where its adversarial patch must simultaneously achieve both removal and fabrication of the bounding box. Also, they did not evaluate the attack's transferability to more advanced MOT algorithms and the effect of digital perturbation on the ReID feature. Therefore, the effectiveness of this attack in more practical settings remains unclear.

Muller et al. [36] introduced AttrackZone, an online and physical tracker hijacking attack against Siamese trackers [37] that exploits the heatmap generation process of Siamese Region Proposal Networks to take control of an object's bounding box. The attack utilizes projectors to present adversarial noise in physically dark environments, which limits its effectiveness in more general scenarios. Although being an online and robust attack against OT, AttrackZone only applies to specific SOT algorithms capable of tracking one target object, and its real-world impact on surveillance systems and autonomous vehicles that rely on MOT algorithms remains open.

There are two notable identity switching attacks in MOT. Lin et al. [13] introduced the tracklet-switch attack against two specific trackers: FairMOT [38] and ByteTrack [8]. They

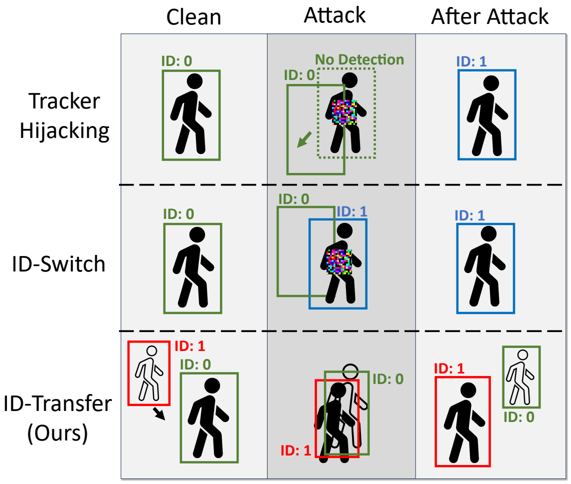


Figure 3: Comparison of existing MOT attacks (by attacking OD modules in digital space) and our ID-Transfer attack (by adversarial trajectories in physical space).

proposed a method to generate digital perturbations to make MOT algorithms confuse intersecting pedestrian trajectories in an offline setting. Although this attack also targets pedestrian tracking scenarios, it is similar to the tracker hijacking attack [11] since it aims to switch the ID assignment in one of the two pedestrians to a different value (not necessarily preserving any original IDs). In addition, they assumed that the attacker can arbitrarily manipulate the captured video frame to add perturbations that affect the bounding box positions and ReID features of multiple objects at the same time. Such a strong threat model limits the practicality of this attack.

Another work introduced the False-Positive-and-False-Negative attack [14] that creates a new identity for a targeted object by erasing the true bounding box while fabricating multiple similar-sized bounding boxes around it. The fake detections, assigned with identities different from the original, will form random trajectories around the true object. After the attack, when the true bounding box detection is restored, it is assigned to one of the new trackers initiated for the fake bounding boxes, resulting in a switch in identity. However, randomly fabricated bounding boxes can cause spatio-temporal inconsistencies that may be detected by existing consistency-based defense methods (e.g., [39, 40]). Furthermore, the attack only works in the digital domain and is shown to be ineffective against advanced MOT algorithms with ReID models enabled [14].

All of these attacks rely on the successful execution of digital perturbations against specific OD models to manipulate the detected bounding boxes (e.g., suppression and fabrication). Therefore, established defenses against OD attacks [41–44] can also mitigate the existing MOT attacks. Meanwhile, the offline setting for these attacks and the dependence on suitable samples also raise questions about their utility in the physical domain. Inspired by adversarial trajectory/maneuver attacks against AD systems [45, 46], which explore subtle manipulation of a vehicle's movement pattern to compromise an AD's trajectory prediction or decision-making process, our ID-Transfer attack against MOT systems leverages physically realizable trajectory generated in an online manner without fooling the perception module. The input to the perception module is unaltered, eliminating the need to attack the OD model or add adversarial perturbations to the input image stream. Thus, our attack fundamentally evades existing OD defenses against perturbation-based attacks. Figure 3 visually compares the different attack goals and methods between existing MOT attacks and our ID-Transfer attack.

3. Problem Statement

3.1. System Model

We consider a real-time tracking-by-detection MOT system that is deployed in pedestrian/vehicle surveillance [1, 5] or autonomous driving (AD) [1] systems for perception. At each time step, the MOT algorithm takes as input a detection state vector $\mathbf{d}_i = (x_1, y_1, x_2, y_2)_i$ representing the upper-left and lower-right corner of a bounding box, for each object $O_i \in \{O_1, ..., O_n\}$ from an object detection (OD) module. The OD module takes input from an RGB camera and is assumed to be capable of providing accurate bounding box prediction for each object of interest in the scene. This assumption isolates the MOT system's ID assignment results to be solely dependent on the object trajectories (which we study in this work) where any potential changes in ID assignment results shall not be attributed to detection errors (e.g., missing detection due to occlusions).

The MOT system maintains a pool of trackers $\mathcal{T} = \{T_1, ..., T_n\}$ for tracked objects. Each tracker keeps track of the associated object states $\mathbf{x} = (u, v, s, r, \vec{u}, \vec{v}, \vec{s})^T \in \mathcal{X}$, where $\vec{\cdot}$ denotes the first-order derivative, and $(u, v), s, r$ are the bounding box center position, scale and aspect ratio, respectively. The tracker is capable of predicting the states $\hat{\mathbf{x}}^t_{\text{prior}}$ of the associated object for the current time step t, which is commonly achieved using the Kalman Filter (KF) [10]. Each tracker's KF makes such predictions by applying a state transition matrix $\mathbf{F}$ to the previous state estimates assuming linear movement between each time step [15]:

$$\hat{\mathbf{x}}^t_{\text{prior}} = \mathbf{F}\hat{\mathbf{x}}^{t-1}_{\text{posterior}} \tag{1}$$

The predicted states of the trackers are used to compare with the OD results for the association. This is achieved by calculating distance metrics based on Intersection-over-Union (IoU) or its variants $d : \mathbb{R}^4 \times \mathbb{R}^4 \rightarrow [0, 1]$ between the detected and predicted bounding boxes for each detection-tracker pair (i, j) to form the distance matrix [47]. The MOT system then solves the weighted bipartite matching problem between existing trackers $T_j \in \{T_1, ..., T_m\}$ and object detections $\mathbf{d}_i \in \{\mathbf{d}_1, ..., \mathbf{d}_n\}$ that minimizes the metric sum using the Hungarian Algorithm [48]. We say that the MOT system assigns ID j (tracker T_j) to an object O_i if and only if there is an edge between $(\mathbf{d}_i, T_j)$ as indicated in the assignment matrix. For brevity, we denote $f(O^t_i) = j$ if the detection of object O_i is assigned to the tracker T_j at time t. After the association phase, each object's detection result is used to perform the KF state update step on the tracker to which it is assigned and obtain the posterior state estimates $\hat{\mathbf{x}}^t_{\text{posterior}}$. The posterior estimates are then used to calculate the prior state estimates for the next time step $t+1$. More details about the matching process can be found in Appendix A.1.

3.2. Problem Formulation

Initially, at $t = 0$, the attacker O_a is assigned to a tracker $T_{\texttt{ID}_\texttt{a}}$ while a targeted object O_b is assigned to a tracker $T_{\texttt{ID}_\texttt{b}}$. The ID-Transfer attack is defined as $f(O_a^t) = \texttt{ID}_\texttt{a}, f(O_b^t) = \texttt{ID}_\texttt{b}, \forall t < \tau$ while $f(O_a^t) \neq \texttt{ID}_\texttt{a}, f(O_b^t) = \texttt{ID}_\texttt{a}, \forall t \geq \tau$. In other words, starting at time $t = \tau$, the attacker switches to a different ID, while its original ID is transferred to the target, as assigned by the MOT system. The attacker aims to craft a series of inputs into the MOT system to achieve this objective.

3.3. Threat Model

We consider an attacker acting as a physical entity that is visible to and tracked by the MOT system. We assume that the attacker and target belong to the same class (e.g., both are pedestrians or vehicles) so that they have comparable sizes. We assume the attacker to be aware of the victim MOT system's camera location and angle, and can observe the trajectory of the target object. The attacker aims to change its assigned MOT ID to a new one while transferring its original ID to the target object tracked by the system. The adversary approaches the above attack goal in the physical world by maneuvering itself along a physically realizable *adversarial trajectory* represented by a series of waypoints (geographical locations in the real world) that will lead to the desired consequences.

The adversary's capability generally fall into two categories: (T1) Optimized adversary (e.g., autonomous vehicles or robots), who possess the capability for real-time calculation and precise motor control, and (T2) Heuristic adversary (e.g., pedestrians or human drivers), who can only perform inexact maneuvers that rely on instinct and heuristics. In addition, we consider adversaries with different knowledge levels of the MOT system:

White-box Attacker. We consider a white-box attacker who has full knowledge of the MOT algorithm. White-box attacks aim to generate optimized trajectories online as the target moves, which requires (T1) adversary.

Black-box Attacker. We consider a black-box attacker with no knowledge of the MOT algorithm being deployed in the system. The attack can be performed by either (T1) or (T2) adversary. The (T1) adversary performs transfer attacks against the system by executing AdvTraj on a surrogate model such as SORT [15], while the (T2) adversary executes adversarial maneuvers based on heuristics.

Both types of attackers are assumed to have realistically limited physical maneuverability. For example, an adversarial walker can only move at reasonable speeds (*e.g.*, 0-3 m/s).

4. AdvTraj: Online ID-Transfer Attack

In this section, we outline the unique challenges of achieving the ID-Transfer attack and our solutions. We further illustrate the conditions necessary to achieve ID-Transfer in the association module of MOT algorithms, which motivates our design of AdvTraj. Our attack pipeline is plug-and-play for tracking-by-detection MOT algorithms that take into account motion information, by employing the corresponding distance metrics into the loss functions.

Furthermore, we summarize the patterns of generated adversarial trajectories under random initial conditions (relative starting positions of the attacker and target) and categorize them into two base cases which other situations can reduce to. Based on this, we develop two highly executable universal adversarial maneuvers that exploit the fundamental vulnerability of the MOT algorithms.

4.1. Challenges

Loss Function Design. The non-differentiability in the matching phase of MOT and limited attacker capabilities raise two design challenges for the loss function. First, the weighted bipartite matching algorithm in the association module of MOT is discrete and hence non-differentiable. Thus, we derive sufficient conditions on the bounding box input to the association module that leads to the desired ID assignments, and perform optimizations with respect to the intermediate results of MOT consisting of the OD detections and KF predictions. Second, our threat model assumes the attacker can only control its own physical movement without the capability to tamper with other objects' detection or motion prediction. It is challenging for the attacker to "prescribe" its own ID to the target under this limited capability assumption, since bipartite matching optimizes for the *sum of distance metrics*. Specifically, only guiding the attacker tracker's KF-predicted states to match the target's detection does not warrant ID-Transfer since the sum of distance metrics for correct ID assignment can still dominate. Thus, we design the adversarial loss function to achieve two objectives simultaneously: match the attacker's OD detection to the target's tracker prediction and match the attacker's tracker prediction to the target's OD detection, while the attacker only controls its own movement.

Manipulating Detected Bounding Boxes. The attacker needs to effectively manipulate the detected bounding box sequence to affect the matching algorithm output in a controlled manner. Contrary to existing MOT attacks that focused on attacking the OD module, which have been shown to be non-robust and model-specific [12], our approach uses physical movement to create an *genuine but adversarial* bounding box sequence. This strategy ensures robustness and independence from specific OD models and also evades OD defenses. Thus, we design AdvTraj to iteratively optimize the attacker's states for each time step that represents the best effort towards the ID-Transfer objective, which also requires less computation resources compared to optimizing over an entire trajectory.

Adherence to Physical Constraints. The attacker must ensure that the desired input of the bounding box to the association phase can be produced by placing itself at physically realizable positions (e.g., not in the sky). This is needed for the attack's practicality and ability to evade anomaly detections. However, enforcing physical constraints on a 2D bounding box sequence is complex, inefficient, and requires extensive knowledge of specific scene topologies. Therefore, instead of optimizing 2D bounding boxes and then translating pixel coordinates into physical locations, we leverage the knowledge of camera parameters[1] to create a differentiable mapping between the physical 3D coordinates[2] and the 2D bounding boxes perceived by the system. This enables us to *directly optimize over intra-frame physical center displacement* for the attacker to represent the desired movement. Physical constraints can thus be incorporated by clipping the movement to be within a realizable range.

4.2. Attack Methodology

We begin by reviewing the MOT association module to identify the conditions required for ID-Transfer. The MOT association module links detected objects (bounding boxes) to existing trackers (predicting current bounding box for associated objects) using a weighted bipartite matching algorithm (e.g., Hungarian Algorithm [48]) that minimizes the *sum of distance metrics* such as Intersection-over-Union (IoU) or its variants [47] $d : \mathbb{R}^4 \times \mathbb{R}^4 \to [0, 1]$, for all combinations of detected and tracker predicted bounding boxes. The resulting bipartite matching represents assigning unique tracker IDs to detections. Thus, given the set of all existing trackers $\mathcal{T} \supset \{T_{\mathtt{ID}_\mathtt{a}}, T_{\mathtt{ID}_\mathtt{b}}\}$ and detected objects $\mathcal{O} \supset \{O_a, O_b\}$, the following set of conditions implies ID-Transfer between O_a and O_b such that $f(O_a) = \mathtt{ID}_\mathtt{b} \wedge f(O_b) = \mathtt{ID}_\mathtt{a}$[3]:

$$d(\mathbf{x}_a, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{b}}) + d(\mathbf{x}_b, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{a}}) < d(\mathbf{x}_a, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{a}}) + d(\mathbf{x}_b, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{b}}) \qquad (\mathtt{C1}),$$

$$d(\mathbf{x}_b, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{a}}) < d(\mathbf{x}_i, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{a}}), \forall i : O_i \in \mathcal{O} \setminus \{O_a, O_b\} \qquad (\mathtt{C2}),$$

$$d(\mathbf{x}_a, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{b}}) < d(\mathbf{x}_i, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{b}}), \forall i : O_i \in \mathcal{O} \setminus \{O_a, O_b\} \qquad (\mathtt{C3}).$$

C1 promotes exchanged IDs between O_a, O_b, whereas C2 and C3 jointly guarantee that no detected objects other than O_a, O_b will be assigned to trackers T_a, T_b. Under our realistic threat model that the attacker can only control its own movement, C2 and C3 can be relaxed in common cases where all tracked entities other than the attacker are benign and have movement patterns consistent with the system's motion prediction model. Thus, to greedily induce ID-Transfer between O_a and O_b, the attacker needs to achieve C1, which represents a necessary but almost sufficient condition for the objective. Notice that, to cause incorrect ID assignments, it is not sufficient for the attacker to solely minimize $d(\mathbf{x}_a, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{b}})$ by placing itself close to the target tracker's predicted states because the matching algorithm optimizes for the *sum* of distance metrics. Therefore, the attacker also needs to maneuver along a trajectory that leads its own tracker predictions close the target object at the same time, i.e., minimizing $d(\mathbf{x}_b, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{a}})$.

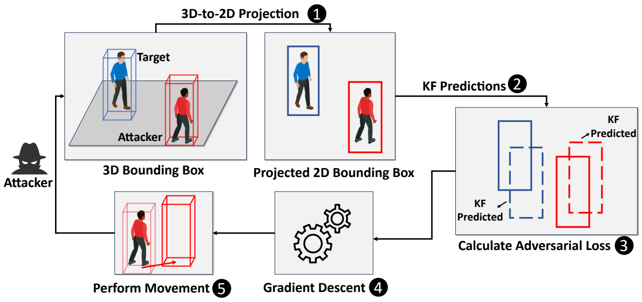


Figure 4: Illustration of AdvTraj's stages.

Figure 4 presents the stages of AdvTraj, where the attacker iteratively finds a physical location to move towards at each time step through optimizing an adversarial loss function designed to encourage C1 and hence inducing ID-Transfer. Specifically, the attacker aims to optimize for a real-world location (x_a^t, y_a^t) so that the system perceived states of the attacker minimizes the distance metrics of the transferred ID assignments: $d(\mathbf{x}_a, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{b}}) + d(\mathbf{x}_v, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{a}})$. In other words, AdvTraj solves:

$$\begin{aligned} &\arg \min_{(x_a^t, y_a^t)} \; d(\mathbf{x}_a, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{b}}) + d(\mathbf{x}_b, \hat{\mathbf{x}}_{\mathtt{ID}_\mathtt{a}}) \\ &\text{subject to } \|(x_a^t, y_a^t) - (x_a^{t-1}, y_a^{t-1})\| \leq \epsilon. \end{aligned} \qquad (2)$$

The constraint represents the limited physical maneuverability of the attacker such that its movement across frames is bounded within a maximum center displacement value ϵ.

Note that there is a gap between the desired real-world coordinates (x_a^t, y_a^t) representing the attacker's location on the physical ground plane and the detection results $\mathbf{d} = (u_1, v_1, u_2, v_2)$ (the upper-left and bottom-right corners) in the image 2D space fed into the MOT algorithm. Therefore, to directly optimize for physical coordinates, the attacker creates a differentiable mapping between the 3D bounding box vertices in the physical world with its 2D bounding box measurement in the image plane, by performing world-to-image projection of its 3D bounding box vertices (➊). In other words, given $\mathbf{S} = \{(x_i, y_i, z_i)^T | 1 \leq i \leq 8\} \in \mathbb{R}^{3 \times 8}$, the set of eight corners of the attacker's 3D bounding box, and the camera parameters (i.e., projection matrix $\mathbf{P}$), the attacker can obtain its 2D bounding box by a 3D-to-2D

[1] The camera parameters are specified by the projection matrix, which consists of the extrinsic and intrinsic matrices. The former relates to the camera's location and angle and the latter is affected only by the camera's internal configurations such as the lens. They can be obtained by performing the standard camera calibration process [49].

[2] An attacker can obtain such information by using a drone-mounted stereo camera or LiDAR to estimate the 3D bounding boxes [36].

[3] For convenience, denote the distance metric between objects i, j: $d(i, j) \equiv d(\mathbf{x}_i, \mathbf{x}_j) := d\Big((u_i, v_i, s_i, r_i), (u_j, v_j, s_j, r_j)\Big)$, where $\mathbf{x} = (u, v, s, r, \vec{u}, \vec{v}, \vec{s})^T$.

projection $\mathcal{P} : \mathbb{R}^{3\times 8} \to \mathbb{R}^4$ such that:

$$\begin{aligned} \mathbf{d} &= \mathcal{P}(\mathbf{S} = (x_i, y_i, z_i)^T | 1 \le i \le 8) \\ &= \left(\min_{1\le i\le 8} x'_i/z'_i, \min_{1\le i\le 8} y'_i/z'_i, \max_{1\le i\le 8} x'_i/z'_i, \max_{1\le i\le 8} y'_i/z'_i \right) \\ \text{where} & \begin{pmatrix} x'_1 & \cdots & x'_8 \\ y'_1 & \cdots & y'_8 \\ z'_1 & \cdots & z'_8 \end{pmatrix} = \mathbf{P} \begin{pmatrix} x_1 & \cdots & x_8 \\ y_1 & \cdots & y_8 \\ z_1 & \cdots & y_8 \\ 1 & \cdots & 1 \end{pmatrix} = \mathbf{P} \begin{pmatrix} \mathbf{S} \\ \mathbf{1} \end{pmatrix}. \end{aligned} \tag{3}$$

[4]

For brevity, we denote $\delta = (\Delta x, \Delta y)$ as the physical center displacement of the attacker along the ground plane representing the applied movement, where $\mathbf{S} + \delta := \{(x_i + \Delta x, y_i + \Delta y, z_i)|1 \le i \le 8\}$ is the 3D bounding box after movement, assuming that the physical height of the object stays constant (❺). Thus far, to achieve ID-Transfer with the target, the attacker finds its desired location to move towards at each time step t by performing gradient descent on the center displacement δ with respect to an adversarial loss (❸-❹):

$$\begin{aligned} &\mathcal{L}(\mathcal{P}(\mathbf{S}^{t-1} + \delta)) = d(g(\mathbf{d}^t), \hat{\mathbf{x}}_b^t) + d(\hat{\mathbf{x}}_b^{t+1}, \hat{\mathbf{x}}^{t+1}) \\ &\text{subject to } \|\delta\| \le \epsilon \end{aligned} \tag{4}$$

where (1) $g : \mathbb{R}^4 \to \mathbb{R}^4$ is the bijective mapping from the two-corner notation to the center-scale-ratio notation for bounding boxes, (2) $\mathbf{d}^t = \mathcal{P}(\mathbf{S}^{t-1} + \delta)$ is the attacker's projected 2D bounding box corresponding to physical movement $\delta = (\Delta x, \Delta y)$, (3) $\hat{\mathbf{x}}_b^{t+1} = \mathbf{F}\hat{\mathbf{x}}_b^t$ is the target's prior state estimates for $t+1$, and (4) $\hat{\mathbf{x}}^{t+1} = \mathbf{F}[\texttt{KF}_{\texttt{ID}_a}.\texttt{update}(\mathbf{x}^t)]$ is the product between the transition matrix and the posterior estimation after update (i.e., the attacker's KF prior predicted states if updated by state observation $g^{-1}(\mathbf{x}^t)$). Note that this single-step KF prediction is linear [10] hence differentiable (❷).

With the adversarial loss function defined, the attacker performs gradient descent (using readily available optimizers like Adam [50]) on δ, which represents the desired movement on the ground plane, clips the result to be within physically realizable regions, and maneuvers towards the target waypoint. The series of waypoints form an adversarial trajectory that encourages ID-Transfer. The complete AdvTraj attack process is summarized in Algorithm 1 and the intuitive explanation of the loss function is presented in Appendix A.2.

4.3. Universal Adversarial Maneuvers

Although AdvTraj automatically generates adversarial trajectories online, it may be challenging for non-automated agents (e.g., humans) to calculate and physically follow the crafted trajectory in real-time, since it would require fine-grained motor control. Thus, based on the understanding of the vulnerability in MOT algorithms that are exploited by AdvTraj, we propose universal adversarial maneuvers (UAMs) that are practically executable by human walkers/drivers.

To achieve this, we start by abstracting the adversarial trajectory generation process, and then investigate the patterns of the generated trajectories (in the 2D plane) to find real-world applicable maneuvers. Although AdvTraj functions regardless of the target's movement, to extract UAMs for common scenarios, we consider a target that moves at a constant velocity (speed and direction), which is the dominant pattern for benign moving objects in daily scenarios. We randomize the starting positions of the attacker relative to the target and perform the attack as the target moves. Detailed analysis and generated trajectories are shown in Appendix A.3.

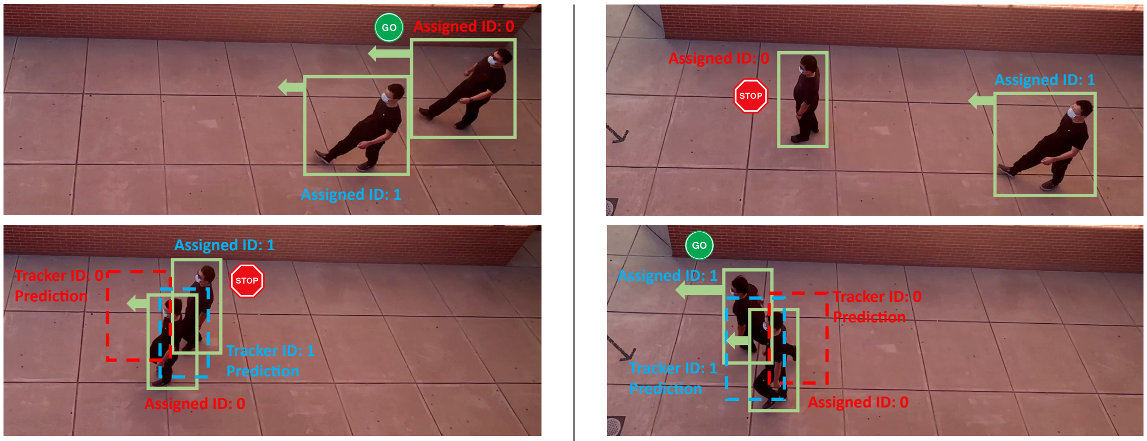
Algorithm 1 AdvTraj ID-Transfer Trajectory Generator

```
Input: Attacker 3D bounding box $\mathbf{S}^{t-1}$, KF trackers $\texttt{KF}_{\texttt{ID}_\texttt{b}}, \texttt{KF}_{\texttt{ID}_\texttt{a}}$,
    maximum displacement $\epsilon$, number of iterations $\texttt{iter}$
Output: Attacker center displacement $\delta$
function AdvTraj-ID-Transfer($\mathbf{S}^{t-1}, \texttt{KF}_{\texttt{ID}_\texttt{b}}, \texttt{KF}_{\texttt{ID}_\texttt{a}}$)
    Initialize $\delta = (0, 0)$
    $\hat{\mathbf{x}}_b^t = \texttt{KF}_{\texttt{ID}_\texttt{b}}.\texttt{predict}()$
    $\hat{\mathbf{x}}_a^t = \texttt{KF}_{\texttt{ID}_\texttt{a}}.\texttt{predict}()$
    if $d(a, \texttt{ID}_\texttt{b}) + d(b, \texttt{ID}_\texttt{a}) > d(a, \texttt{ID}_\texttt{a}) + d(b, \texttt{ID}_\texttt{b})$ then
        $\hat{\mathbf{x}}_b^{t+1} = \mathbf{F}\hat{\mathbf{x}}_b^t$
        Initialize $\texttt{i} = 0$
        while $\texttt{i} < \texttt{iter}$ do
            $\Delta = \nabla_\delta \mathcal{L}(\mathcal{P}(\mathbf{S}^{t-1} + \delta))$
            $\delta = \texttt{GradientDescent}(\delta, \Delta)$
            $\texttt{i} = \texttt{i} + 1$
        end while
        $\delta = \texttt{Clip}(\delta, \epsilon)$
    end if
    return $\delta$
end function
while IDs are not transferred and a new frame arrives do
    $\delta = \texttt{AdvTraj} - \texttt{ID} - \texttt{Transfer}(\mathbf{S}^{t-1}, \texttt{KF}_{\texttt{ID}_\texttt{b}}, \texttt{KF}_{\texttt{ID}_\texttt{a}})$
    $\texttt{PerformMovement}(\delta)$
end while
```

We observe that the generated trajectories share a prominent pattern where the attacker initially attempts to close its distance to the target greedily regardless of the initial position (P1), and performs maneuvers along the same direction as the target with varying speed (non-linear movement) after being around the target's location (P2). Furthermore, these adversarial trajectories can be divided into two categories, where the attacker reaches the target's history path (behind the target) or the target's projected path (ahead of the target), before the ID-Transfer. By isolating these two common cases that other adversarial trajectories can reduce to, we further examine the attacker's specific movement patterns, which can be summarized as follows.

When the attacker starts behind the target, it approaches the target at a higher speed and converges to the same moving direction as the target (P1). As the attacker approaches the target, it abruptly decelerates, positioning itself behind the target (P2). However, the KF prediction, based on the attacker's previous high-speed trajectory, would predict the attacker to be ahead of itself but close to the target. Therefore, the sum of distance metrics becomes lower for the attacker being assigned to the target's tracker and vice versa.

[4] z'_i is the scalar value of 2D homogeneous point representation (with 2 DoF) such that $(x'_i/z'_i, y'_i/z'_i, 1) = (x'_i, y'_i, z'_i)$.



(a) Attacker starts behind the target (Go-and-Stop)

(b) Attacker starts ahead of the target (Stop-and-Go)

Figure 5: Illustration of the two universal adversarial maneuvers for ID-Transfer. Top/lower walker is the attacker/non-cooperating target, respectively. Solid green boxes represent the detection results of the OD model, while dashed bounding boxes represent predictions made by the KF motion prediction module of the MOT system.

For the attacker initiated ahead of the target, it moves towards the target's projected path and moves in the exact opposite direction as the target (due to greedy reduction in distance) (P1). When the target passes by, the attacker turns to move in the same direction as the target but at a higher speed when the target passes (P2). This sudden acceleration also places the attacker's actual position close to the target's KF-predicted location, while its own KF prediction (lagging behind) is close to the target's actual position. This also results in the sum of distance metrics becoming lower for the attacker being assigned to the target's tracker and vice versa.

These two patterns of adversarial trajectories can be summarized into two UAMs, namely *Go-and-Stop* and *Stop-and-Go*, which are visually demonstrated in Figure 5. The attacker starts by closing its distance from the target by moving towards the target's history path (behind) or projected path (ahead), then employs the corresponding tactic to complete the ID-Transfer attack. Both UAMs increase $d(a, \mathtt{ID_a})$ drastically by inducing a large difference between $\mathbf{d}_a$ and $\hat{\mathbf{d}}_a$ when $\hat{\mathbf{d}}_a$ is expected to be close to the actual states of the target object $\mathbf{d}_b$. Since weighted bipartite matching aims to minimize the global assignment cost, misalignment $d(a, \mathtt{ID_a}) + d(b, \mathtt{ID_b}) > d(a, \mathtt{ID_b}) + d(b, \mathtt{ID_a})$ in predicted and actual positions can deceive the system into incorrectly associating the identities.

5. Evaluation

We evaluate AdvTraj by simulating the (T1) optimized adversary (e.g., automated agents) in the open source simulator CARLA [51] and conducting real-world experiments for the (T2) heuristic adversary (e.g., human walkers/drivers) under various application scenarios of MOT systems.

5.1. Evaluation Methodology and Setup

Evaluation Metrics. We measure the effectiveness of AdvTraj by the *attack success rate* (ASR) as the number of successful attacks over the total number of randomized simulations/real-world experiments. For the (T1) optimized adversaries, an attack is considered successful if the IDs of the attacker and the target object assigned by the MOT system are swapped and stay exchanged after the attack, as compared to the initial assignment when their trackers are created. For the (T2) heuristic adversaries, the ASR is calculated as the number of successful attacks over the total number of attacks performed in the whole continuous video recorded for each scenario where the attacker does not have the output from the system when performing the universal adversarial maneuvers (UAMs). Since naturally occurring missed detections during non-line-of-sight occlusions may lead to tracker loss, we consider an attack to be successful if the ID of the attacker is transferred to the target or the target's ID is transferred to the attacker after the attack. This is a stronger definition than the ID-Switch attack used in previous works on attacking MOT [12–14, 34] since we require the preservation of at least one original ID of the attacker or target which is assigned incorrectly due to the attack.

MOT Algorithms. Although different MOT algorithms vary in specific designs (e.g., association distance metrics, estimated states, KF parameters), most state-of-the-art models adopt the tracking-by-detection framework and use KF (or its variant) as the motion prediction module, which can be attacked using AdvTraj by employing the corresponding distance metrics and KF parameters in Equation 4. However, these common design principles also allow the black-box attacker to perform transfer attacks using surrogate models. Thus, to evaluate the effectiveness of white-box attacks (by (T1) adversary), we implement AdvTraj against the representative tracking-by-detection algorithm SORT[15]

TABLE 1: Details of evaluated MOT algorithms.

	Motion Prediction Model	ReID Included	IDF1↑†
SORT [15]	Std. Kalman Filter	N	76.9
OC-SORT [9]		N	77.5
ByteTrack [8]		N††	79.3
Deep OC-SORT [53]		Y	80.6
BoT SORT [22]		Y	80.2
StrongSORT [52]	NSA Kalman Filter	Y	82.3

† Ratio of correctly identified detections over the average of ground truth and predicted detections, evaluated on MOT17 [54].
†† ReID can be incorporated, though it is shown that the best performing model uses IoU only [8], which is also the original implementation by the authors.

by writing the attack module, simulation APIs, the SORT algorithm and its KF dependencies using Python 3.8 and Tensorflow 2.6.0 totaling 2,585 LoC[5].

We evaluate the transferability of the adversarial trajectories generated against SORT to other MOT algorithms to assess the effectiveness of black-box attacks by (`T1`) adversary. Specifically, we consider five other state-of-the-art MOT algorithms (detailed in Table 1): OC-SORT [9], ByteTrack [8], BoT-SORT [22], and StrongSORT [52]. We use the respective default parameters of each algorithm for evaluation. For trackers that require appearance descriptors, we use the same pre-trained OSNET [25] as the ReID backbone for fair comparison.

On the other hand, the (`T2`) adversary performing UAMs do not have the capability to conduct real-time optimization and fine-grained motor control. Thus, we conduct real-world experiments for black-box ID-Transfer attacks by (`T2`) adversary and evaluate the recorded footage on each of the six MOT algorithms.

Scenario Setup. We evaluate AdvTraj in two applications: pedestrian/vehicle surveillance and autonomous driving (AD). These scenarios were chosen to represent the safety-critical applications of MOT systems with the consequences of losing track of a target of interest and unsafe driving decisions when the attack occurs. We primarily focus on the pedestrian surveillance and AD scenarios in this section, where we employ walkers as tracked objects[6]. The setup and results on vehicle surveillance are presented in Appendix A.4.

For the pedestrian surveillance scenario, a fixed camera is mounted at a high position (e.g., edge of a building) as the input sensor to the MOT algorithm. It provides wide views with less occlusion of the open walking area beneath it.

For the AD scenario, the MOT system is deployed as part of the perception module of the AD system, where the image sensor captures video streams as input. We consider two representative patterns for vehicle-pedestrian interactions: (1) pedestrians walk *perpendicular* to a stopped AD, which usually occurs at a crosswalk; (2) pedestrians on the sidewalk walking *parallel* to the AD, which drives forward at a constant speed of 20 KM/h.

Simulation Setup. To simulate the (`T1`) adversary and evaluate the effectiveness of AdvTraj, we leverage the CARLA simulator, known for its physics-compliant engine and photorealistic rendering. The camera parameters, captured video frames, 3D coordinates, and bounding boxes of detected objects can be extracted directly from CARLA. We choose appropriate locations in CARLA Town 01/05 and perform 100 randomized simulations for each of the three scenarios: (1) pedestrian surveillance, (2) AD-parallel, and (3) AD-perpendicular. The MOT system retrieves 1920 × 1080 RGB images and object bounding boxes at 10 frames per second from CARLA and performs real-time ID assignments. For each simulation, the spawn locations for the attacker and target are randomly chosen in walkable/drivable areas in the scene. Also, to understand how the target movement speed affects the effectiveness of AdvTraj, for each scenario, we additionally perform 100 randomized simulations for each target movement speed group set between 0.5-2.5 m/s.

We also assess the impact of the different pedestrian appearances on the attack's effectiveness against MOT algorithms with ReID model embedded. For the same set of adversarial trajectories generated by AdvTraj, we vary the appearances of the attacker and target with 25 different combinations of CARLA blueprints (different styles and colors of clothing). Then, the appearance-mutated simulations are replayed to ReID-enabled MOT algorithms to recalculate ASRs under each attacker-target appearance pair. The difference in appearances is calculated using the average cosine distance (same metric used by the evaluated MOT algorithms) between the attacker's and target's ReID feature vectors across all frames. The corresponding results are presented in Appendix A.5.

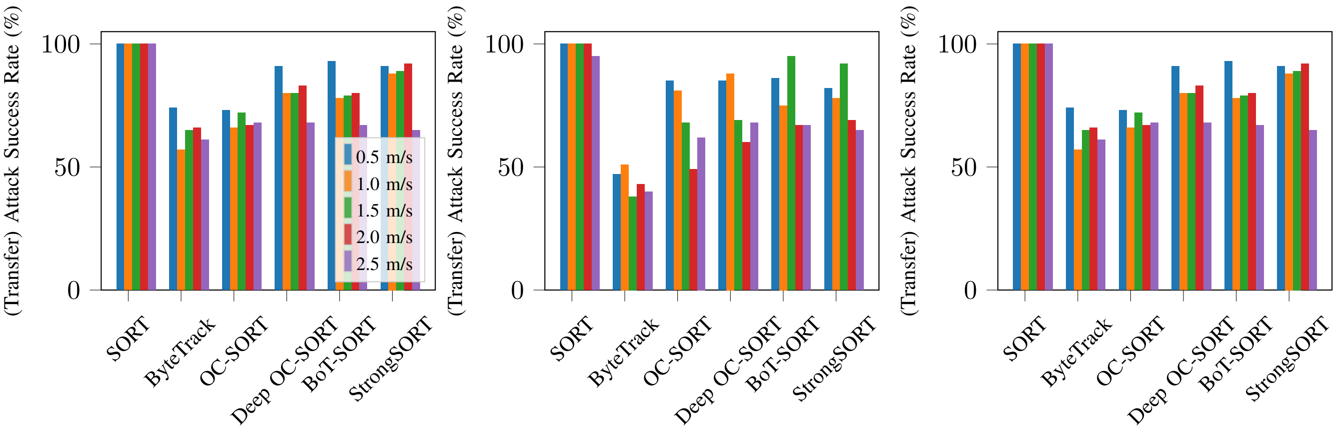
Implementation Details. We ran the simulation on a desktop with i9-13900K CPU and RTX 4090 GPU. We used the Adam optimizer [50] with a learning rate $\alpha = 0.1$, the number of iterations $\texttt{iter} = 5$, and set the maximum center displacement $\epsilon = 0.35$ (attacker walker cannot walk faster than 3.5 m/s).

Real-World Experiment Setup. We conducted real-world experiments to evaluate the attack effectiveness for (`T2`) adversaries. In both scenarios, the attacker only observes the positions of the MOT system's image sensor and the target walker, without knowledge of the deployed MOT system nor the ability for real-time optimization and precise motor controls. The image sensor captures 1920 × 1080 videos at 30 frames per second, while YOLOv5 [55] is used as the OD model. Wearing similar/distinct outfits than the target (black-black/black-white), the attacker walker uses one of the two black-box ID-Transfer UAMs explained in Section 4.3 depending on the relative starting position (Go-and-Stop if behind the target, Stop-and-Go if ahead of the target).

For pedestrian surveillance, a fixed RGB sensor is mounted on the rooftop acting as the surveillance camera monitoring an outdoor terrace. The target pedestrian randomly picks a starting position and walks straight forward at a constant speed of around 1.1 M/s. We conducted the experiments 40 times for each UAM.

For the AD application, we emulate the perceived video stream of a vehicle by mounting an RGB sensor on the front

[5] Available at https://github.com/ch3ny1/AdvTraj_ID_Transfer.
[6] Agents in CARLA simulator are available as either walkers or vehicles.



(a) Pedestrian Surveillance

(b) AD-Perpendicular

(c) AD-Parallel

Figure 6: White-box attack success rate on SORT and black-box transfer attack success rates on other MOT algorithms of AdvTraj generated adversarial trajectories with respect to different target movement speeds.

TABLE 2: AdvTraj white-box and black-box attack success rates in randomized simulation on different MOT models by (T1) optimized adversary, compared with baseline ID misassignment rates when no attack is performed.

	Pedestrian Surveillance	**AD-Perpendicular**	**AD-Parallel**
White-Box	Attack Success Rate / Baseline		
SORT	100% / 0%	100% / 15%	100% / 6%
Black-Box	Transfer Attack Success Rate / Baseline		
ByteTrack	89% / 0%	45% / 16%	66% / 5%
OC-SORT	92% / 0%	69% / 15%	69% / 3%
Deep OC-SORT	93% / 0%	74% / 15%	80% / 4%
BoT-SORT	74% / 0%	78% / 12%	79% / 10%
Strong SORT	84% / 0%	77% / 6%	85% / 2%

hood of a Toyota Prius sedan to record footage for offline analysis. For each of the parallel and perpendicular scenarios, we perform the attack 40 times. The experiments are carried out safely in a large empty parking lot.[7]

Baseline. AdvTraj poses a novel threat that has not been explored before, where the ID-Transfer differs from previous works in both attacker goal (confuse two tracked objects instead of one) and attack methodology (using physical adversarial trajectory instead of attacking OD). Therefore, to better quantify the effectiveness of our attack, we collect baseline cases for each scenario where two pedestrians have random intersecting trajectories (different speeds, directions, movement patterns etc). Specifically, for each of the pedestrian surveillance, AD-perpendicular, and AD-parallel, we collect 100 samples in CARLA (where the attacker and target have the exact same appearance for estimating an upper bound of ID misassignment rate) and 100 samples in real-world (50 wearing black/black outfit and 50 wearing black/white outfit). We evaluate the rate of ID misassignment (ID-Transfer in either direction) under these baseline cases.

5.2. Simulation Results

Attack Effectiveness and Transferability. Table 2 details the baseline ID misassignment rates, white-box ASR on SORT and black-box transfer ASR on other MOT algorithms, where the attacker and target share the same appearance.

The white-box attack on SORT achieved a 100% success rate across all scenarios, emphasizing its complete vulnerability when full system knowledge is available. The black-box transfer attacks on other MOT algorithms show varying degrees of success. Notably, ByteTrack and OC-SORT show higher resilience to the transfer attack compared to the other three ReID-enabled MOT algorithms. This suggests similar appearances amplify the ID-Transfer potential for these models where their performance can degrade to be worse than motion-only MOT algorithms. Nevertheless, the transfer ASRs surpass all evaluated algorithms' baseline ID misassignment rates. This implies that even when the attacker lacks complete knowledge, the system could still face threats from black-box transfer attacks.

Impact of Different Target Speeds. Figure 6 shows the white-box ASRs on SORT and black-box ASRs on other MOT algorithms where the attacker and target have the same appearance but the target movement speeds are set differently. Note that the ASRs on the white-box victim algorithm SORT almost remain 100% for different target movement speeds, except dropping slightly when the target moves at 2.5 m/s in the AD-perpendicular case. Although the white-box AdvTraj consistently generates adversarial trajectories before successful ID-Transfer, a few unsuccessful scenarios in AD-perpendicular happened when the attacker randomly spawned at locations too far from the target to be able to close the distance within the maximum simulation duration set at 150 frames. On the other hand, the black-

[7] Black-Box AdvTraj demo: https://youtu.be/ETuJQFlxqIU

box ASR on the other MOT algorithms generally decreases, although mildly, as the target movement speed increases from 0.5 to 2.5 m/s.

5.3. Real-World Experiment Results

Attack Effectiveness. The real-world experiment results summarized in Table 3 demonstrate the effectiveness of UAMs performed by (T2) adversaries against various MOT algorithms under black-box settings.

StrongSORT demonstrates high robustness to the UAMs except for the AD-parallel scenario when the attacker and target have similar appearances, while other MOT algorithms are generally more susceptible in the AD-perpendicular scenario. This is because the horizontal placement of RGB sensor in this case enables the attacker to have a larger overlap with the target's bounding box (hence more flexibility in optimizing the adversarial objective). For pedestrian surveillance and AD-parallel scenarios, the vulnerabilities vary for different MOT algorithms. For example, ByteTrack is more susceptible in the pedestrian surveillance case, whereas BoT-SORT and StrongSORT are more vulnerable to AD-parallel when the two walkers' appearances are similar.

Note that the visual distinction between the attacker and target helps alleviate the attack's efficacy, as expected, in Deep OC-SORT, BoT-SORT, and StrongSORT, which have the ReID model enabled. Although BoT-SORT and SrongSORT demonstrate higher robustness in the distinct appearance cases, Deep OC-SORT still falls susceptible to the attack with ASR greater than 20%. This suggests that the appearance matching and/or update mechanisms in Deep OC-SORT are less robust compared to BoT-SORT and StrongSORT. StrongSORT's high resilience against the adversarial trajectories in real-world experiments reflects that its incorporation of the NSA Kalman Filter (with adaptive covariance parameters) and non-linear Gaussian-smoothing interpolation during short-term occlusions may help mitigate the effect from UAMs by non-automated agents.

On the other hand, since SORT, ByteTrack, and OC-SORT do not consider appearance but only use motion information for detection-tracker association, the differences in ASR between experiments with similar/different appearances are small and can be attributed to natural variations in real-world experiments.

6. Discussion

Limitations. From the evaluation results, there remains a gap in attack success rates (ASR) between the simulation and real-world experiments due to different capability levels of the (T1) optimized and (T2) heuristic adversaries. A successful heuristic-based attack in the physical world relies on the attack performer's dexterity and instinct in executing the maneuver. Therefore, the real-world experiments for universal adversarial maneuvers mainly serve as a proof-of-concept, where the attack success rates are calculated on the whole unedited footage to reflect the practicality of the attack. To empower a human attacker as an optimized adversary in practice, one may consider using augmented reality technology, which provides real-time localization and calculation capability, and overlays on the environment for trajectory guidance [56].

TABLE 3: AdvTraj black-box attack success rates in real-world experiments by (T2) heuristic adversary, compared with baseline ID misassignment rates without attack.

	Pedestrian Surveillance	**AD-Perpendicular**	**AD-Parallel**
Similar Appearance - Attack Success Rate / Baseline			
SORT	30% / 8%	37.5% / 16%	30% / 15%
ByteTrack	40% / 12%	37.5% / 16%	20% / 10%
OC-SORT	27.5% / 6%	35% / 10%	40% / 5%
Deep OC-SORT	45% / 14%	30% / 12%	45% / 20%
BoT-SORT	6.5% / 4%	37.5% / 10%	20% / 5%
StrongSORT	0% / 0%	0% / 0%	15% / 0%
Distinct Appearance - Attack Success Rate / Baseline			
SORT	32.5% / 6%	37.5% / 16%	25% / 15%
ByteTrack	30% / 8%	32.5% / 16%	20% / 5%
OC-SORT	32.5% / 6%	40% / 12%	30% / 5%
Deep OC-SORT	22.5% / 10%	32.5% / 8%	20% / 10%
BoT-SORT	5% / 2%	17.5% / 2%	0% / 0%
StrongSORT	0% / 0%	0% / 0%	0% / 0%

Another cause for the gap in ASR lies in the fact that the bounding boxes produced in the simulation are accurate, where the results of ID assignments can be attributed to the bounding-box trajectories. However, in real-world experiments, non-line-of-sight situations prevent the OD model from producing accurate bounding boxes for the (partially/completely) occluded object. To alleviate such impact, our real-world experiments were performed where the attacker was closer to the camera (blocks the target) for half of the total trials and was further away from the camera (blocked by the target) for the other half.

Attacker-Target Interaction. In uncontrolled real-world scenarios, where the target could potentially react to the attacker's actions, the attacker should carefully execute maneuvers to minimize suspicion. The black-box heuristics described in our paper are foundational for inducing ID-Transfer in MOT. However, to be more stealthy, the attacker can employ practical tricks such as running toward a walking pedestrian and then stopping to pick up a dropped wallet, or turning back as if they forgot something. Similarly, using the stop-and-go technique, while appearing unrelated to the target, can subtly influence the trajectory of the target without arousing suspicion. The key for the attacker is to perform these maneuvers naturally, reducing target reaction and thus maintaining the effectiveness of the attack.

Ethical Considerations. To ensure ethical integrity, we took several precautions in designing and conducting our real-world experiments. Our experiments did not involve any participation from individuals outside the research team, and no identifiable private information was studied, analyzed, or stored. The experimental locations were carefully selected to be in safe, public spaces with minimal foot traffic, and any third parties who happened to be nearby were far from the experiment site and remained unidentifiable.

Countermeasures. MOT systems with a single image sensor input work in the 2D plane and lack the capability to distinguish objects utilizing depth information, which may result in confusion when two detected bounding boxes overlap. Thus, incorporating robust depth information, such as through multi-sensor fusion with LiDAR or multiple cameras, could be a potential defense method. In addition, the motion prediction module of the MOT systems may be trained to adapt to adversarial trajectories against black-box attackers, specifically by adjusting the system parameters to better fit adversarial movement patterns. However, this approach incurs a trade-off between adversarial robustness and benign performance. MOT systems are typically configured for high tracking accuracy in benign scenarios, with parameters optimized to align with common movement patterns and minimize the distance between object detection results and predicted locations. Adjusting these parameters to enhance robustness against adversarial attacks may compromise the system's performance in normal situations, potentially causing frequent ID switches and other tracking errors when the system is not under attack. Further investigating this trade-off will be a future research direction.

Nevertheless, the linear motion assumption made by various SORT-like algorithms is inherent to the use of standard Kalman Filter as the motion prediction model. Although it strikes a balance between real-time performance and accuracy in the benign case, more expressive motion prediction models such as the Extended Kalman Filter [57] and the Particle Filter [58] can be adopted to address potential threats introduced by adversarial trajectories. Although StrongSORT has the highest computation overhead and slowest tracking speed among the evaluated MOT algorithms, its non-linear Gaussian-Smoothed Interpolation and NSA Kalman Filter are shown to make the model robust against the black-box UAMs in our real-world experiments against (`T2`) adversary.

The incorporation of ReID features as part of the association distance metrics, though introducing extra computation overhead, also helps dilute the contribution of motion information on matching decisions hence mitigating the attack. However, the ReID feature requires robust update and matching mechanisms to improve the MOT algorithms' overall robustness, since the bounding box of one object may contain information for another object when they overlap. This can also explain the performance gap between simulation and real-world experiments for ReID-enabled MOT algorithms, where the accurate bounding boxes for overlapping objects can result in confusion of their appearance (in simulation). Notably, a popular MOT algorithm DeepSORT [18] uses appearance information only as the primary metric to perform the association (the motion information is only used to filter out unlikely matches), which is susceptible to ID-Transfer even when two tracked objects overlap in the benign case. Based on our real-world experiments, DeepSORT has 22.5%-30% ID-Transfer rates in baseline cases with distinct appearances and 35%-40% with similar appearances, and the black-box attack success rates range from 50%-75%. Although MOT algorithms can use both appearance and motion information to achieve better performance in benign scenarios, it still leaves the door open for the attacker to impact the system using adversarial trajectories, as motion prediction is essential in MOT to consistently track objects especially when appearance/biometric data is unavailable or inapplicable.

7. Conclusions

We introduce AdvTraj, a novel physical ID-Transfer attack against MOT systems using adversarial trajectories. AdvTraj exploits the vulnerabilities in the association phase commonly found in various state-of-the-art MOT algorithms for online trajectory generation, eliminating the need to attack the object detection model. We simulated the optimized attacker in CARLA and performed real-world experiments for the black-box heuristic attacker with application scenarios in pedestrian/vehicle surveillance and autonomous vehicles. We demonstrated the transferability of the attack across different MOT algorithms and evaluated the impact of appearance/speed differences between the attacker and the target. In the simulation, AdvTraj produces attack success rates of up to 100% for the white/black-box attackers, respectively. Our proposed universal adversarial maneuvers can achieve up to 45% attack success rates by human performers in the real world.

Future work will expand our analysis to realize the white-box attack in the real world by relaxing the assumption that the attacker's surrogate OD model can always produce accurate bounding-box predictions and empowering the experimental subject to be an optimized attacker. In addition, we plan to extend our study to explore the impact of execution errors of adversarial trajectories on attack success rates and the possibility of other forms of ID-manipulation attacks and their end impact on autonomous systems, such as missed detection or bounding box merging, which often occur in current OD models when two objects are close in distance. Lastly, we plan to develop defenses against ID-manipulation attacks without sacrificing benign performance and ultimately develop a standardized evaluation framework for MOT system robustness against physical attacks.

Acknowledgment

Chenyi Wang would like to thank the Herbold Foundation for the Herbold Fellowship, which partly supported his work. The work of Chenyi Wang and Ming Li was supported in part by FY24 and FY25 Eighteenth Mile TRIF funding from the University of Arizona. Chenyi Wang, Ming Li, and Ryan Gerdes were supported in part by the U.S. Army Research Office (ARO) under grant W911NF-21-1-0320. The work of Z. Berkay Celik was supported in part by the National Science Foundation (NSF) under grant CNS-2144645 and grant IIS-2229876. We also thank our lab members: Jingcheng Li, Zhiwu Guo, and Ziqi Xu for helping with the real-world experiments.

References

[1] B. A. Team, *Apollo: Open source autonomous driving*, https://github.com/ApolloAuto/apollo, 2017.

[2] W. LLC, *Waymo*, https://waymo.com.

[3] comma.ai, *OpenPilot*, https://github.com/commaai/openpilot.

[4] S. Kato *et al.*, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *International Conference on Cyber-Physical Systems (ICCPS)*, 2018.

[5] M. Elhoseny, "Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems," *Circuits, Systems, and Signal Processing*, 2020.

[6] X. Luo, R. Zhao, and X. Gao, "Research on UAV multi-object tracking based on deep learning," in *International Conference on Networking, Sensing and Control (ICNSC)*, 2021.

[7] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs)," in *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[8] Y. Zhang *et al.*, "ByteTrack: Multi-object tracking by associating every detection box," *CoRR*, 2021.

[9] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-Centric SORT: Rethinking SORT for robust multi-object tracking," in *CVPR*, 2023.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," in *ASME Journal of Basic Engineering*, 1960.

[11] Y. Jia *et al.*, "Fooling detection alone is not enough: Adversarial attack against multiple object tracking," in *ICLR*, 2020.

[12] C. Ma, N. Wang, Q. A. Chen, and C. Shen, "WIP: Towards the practicality of the adversarial attack on object tracking in autonomous driving," *ISOC Symposium on Vehicle Security and Privacy*, 2023.

[13] D. Lin, Q. Chen, C. Zhou, and K. He, "TraSw: Tracklet-switch adversarial attacks against multi-object tracking," *CoRR*, 2021.

[14] T. Zhou, Q. Ye, W. Luo, K. Zhang, Z. Shi, and J. Chen, "F&F attack: Adversarial attack against multiple object trackers by inducing false negatives and false positives," in *ICCV*, 2023.

[15] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *International Conference on Image Processing (ICIP)*, 2016.

[16] H. Fan *et al.*, "Lasot: A high-quality benchmark for large-scale single object tracking," in *CVPR*, 2019.

[17] M. Müller, A. Bibi, S. Giancola, S. Al-Subaihi, and B. Ghanem, "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild," in *ECCV*, 2018.

[18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *International Conference on Image Processing (ICIP)*, 2017.

[19] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," *ECCV*, 2020.

[20] Z. Lu, V. Rathod, R. Votel, and J. Huang, "RetinaTrack: Online single stage joint detection and tracking," in *CVPR*, 2020.

[21] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," in *CVPR*, 2022.

[22] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," *arXiv:2206.14651*, 2022.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[24] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, 2017.

[25] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *ICCV*, 2019.

[26] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *ICLR*, 2014.

[27] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. L. Yuille, "Adversarial examples for semantic segmentation and object detection," in *ICCV*, 2017.

[28] S. Chen, C. Cornelius, J. Martin, and D. H. Chau, "ShapeShifter: Robust physical adversarial attack on Faster R-CNN object detector," in *Machine Learning and Knowledge Discovery in Databases*, 2018.

[29] X. Liu, H. Yang, Z. Liu, L. Song, Y. Chen, and H. Li, "DPATCH: An adversarial patch attack on object detectors," in *AAAI*, 2019.

[30] K. Eykholt *et al.*, "Robust physical-world attacks on deep learning visual classification," in *CVPR*, 2018.

[31] K. H. Chow *et al.*, "Adversarial objectness gradient attacks in real-time object detection systems," in *International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*, 2020.

[32] B. Yan, D. Wang, H. Lu, and X. Yang, "Cooling-shrinking attack: Blinding the tracker with imperceptible noises," in *CVPR*, 2020.

[33] Q. Guo *et al.*, "SPARK: spatial-aware online incremental attack against visual tracking," in *ECCV*, 2020.

[34] X. Yan, X. Chen, Y. Jiang, S. Xia, Y. Zhao, and F. Zheng, "Hijacking tracker: A powerful adversarial attack on visual tracking," in *International Conference on Acoustics, Speech and Signal Processing*, 2020.

[35] L. Ding *et al.*, "Towards universal physical attacks on single object tracking," in *AAAI*, 2021.

[36] R. Muller, Y. Man, Z. B. Celik, M. Li, and R. M. Gerdes, "Physical hijacking attacks against object trackers," in *ACM Conference on Computer and Communications Security (CCS)*, 2022.

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

[38] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, 2021.

[39] Y. Man, R. Muller, M. Li, Z. B. Celik, and R. M. Gerdes, "That person moves like A car: Misclassification attack detection for autonomous systems using spatiotemporal consistency," in *USENIX Security Symposium*, 2023.

[40] R. Muller, Y. Man, R. Gerdes, M. Li, J. Petit, and Z. B. Celik, "Vogues: Validation of object guise using estimated components," in *USENIX Security Symposium*, 2024.

[41] H. Zhang and J. Wang, "Towards adversarially robust object detection," in *ICCV*, 2019.

[42] S. A. A. Shah, M. Bougre, N. Akhtar, M. Bennamoun, and L. Zhang, "Efficient detection of pixel-level adversarial attacks," in *International Conference on Image Processing*, 2020.

[43] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of JPG compression on adversarial images," *CoRR*, 2016.

[44] A. Amirkhani and M. P. Karimi, "Adversarial defenses for object detectors based on Gabor convolutional layers," *Vis. Comput.*, 2022.

[45] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On adversarial robustness of trajectory prediction for autonomous vehicles," in *CVPR*, 2022.

[46] R. Song, M. O. Ozmen, H. Kim, R. Muller, Z. B. Celik, and A. Bianchi, "Discovering adversarial driving maneuvers against autonomous vehicles," in *USENIX Security Symposium*, 2023.

[47] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *AAAI*, 2020.

[48] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, 1955.

[49] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[50] D. P. Kingma and J. Ba, in *ICLR*, 2015.

[51] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Annual Conference on Robot Learning*, 2017.

[52] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, "GIAOTracker: A comprehensive framework for MCMOT with global information and optimizing strategies in VisDrone 2021," in *ICCV Workshops*, 2021.

[53] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification," *arXiv:2302.11813*, 2023.

[54] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831*, 2016.

[55] G. Jocher, *YOLOv5 by Ultralytics*, version 7.0, 2020.

[56] A. B. Craig, "Understanding augmented reality: Concepts and applications," 2013.

[57] G. L. Smith, S. F. Schmidt, and L. A. McGee, *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. National Aeronautics and Space Administration, 1962.

[58] H. R. Künsch, "Particle filters," *Bernoulli*, 2013.

[59] G.-S. Lab, *Carla-apollo bridge*, https://github.com/guardstrikelab/carla_apollo_bridge.

Appendix A.

A.1. Weighted Bipartite Matching in MOT

In MOT, weighted bipartite matching is crucial for associating detected objects with their corresponding trackers across video frames. A bipartite graph consists of two disjoint sets of vertices: one representing detected object bounding boxes and the other representing bounding boxes predicted from previously tracked object trajectories. Each edge between these sets is weighted, based on the cost or distance between a detection and a tracker prediction measured by Intersection-over-Union (or its variants) [47]. The goal is to find a matching that minimizes the total cost to encourage correct ID assignments and consistent tracking. The Hungarian algorithm (a.k.a. Kuhn-Munkres algorithm) [48] is an efficient and widely used method for solving the assignment problem in MOT.

A.2. Details of Loss Function

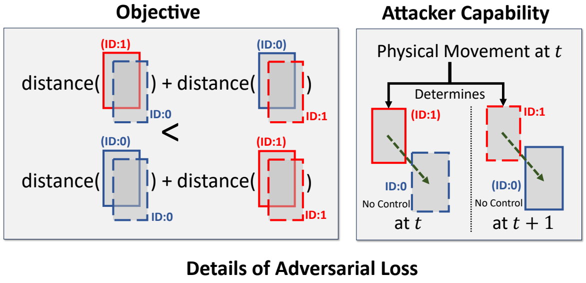


Figure 7: Illustration of the adversarial loss details. ID within parenthesis is the ground truth tracker ID the bounding box belongs to. The attacker's movement from $t-1$ to t (which is optimized via gradient descent) determines its detected bounding box (solid) at t and the KF predicted bounding box (dashed) at $t+1$. The attacker aims to simultaneously push its detected and KF-predicted bounding boxes closer to the target's KF-predicted and detected bounding boxes, respectively.

Intuitively, minimizing Eq. 4 produces the attacker's optimal states that achieve two goals simultaneously: (1) the attacker's current states $\mathbf{d}^t$ being close to the target object's current KF predicted states $\hat{\mathbf{x}}_b^t$, which encourages the attacker to obtain the target's identity (2) the attacker's KF predicted states $\hat{\mathbf{x}}^{t+1}$, if updated by $\mathbf{x}^t$, being close to the target object's KF predicted states $\hat{\mathbf{x}}_b^{t+1}$ for the next time step, which promotes assigning the attacker's ID to the target. Note that the attacker's movement at any time t only affects its detected states $\mathbf{x}^t$ but its corresponding next-step KF predicted states $\hat{\mathbf{x}}^{t+1}$. Nevertheless, optimizing our loss function produces an adversarial trajectory where its history (say $t = 0, ..., \tau - 1$) has encouraged assigning the attacker's KF predicted states to the target's detection at τ. Therefore, these two goals jointly represent the best efforts at each time step of satisfying $\mathtt{C1}$:$d(\mathbf{x}_a, \hat{\mathbf{x}}_{\mathtt{ID_b}}) + d(\mathbf{x}_b, \hat{\mathbf{x}}_{\mathtt{ID_a}}) < d(\mathbf{x}_a, \hat{\mathbf{x}}_{\mathtt{ID_a}}) + d(\mathbf{x}_b, \hat{\mathbf{x}}_{\mathtt{ID_b}})$ under our assumption that the attacker cannot influence the movement or motion prediction of the target object while the ID of the target is correctly assigned.

The Intersection-Over-Union (IoU) association distance metric $(1-\text{IoU})$ used by SORT and various MOT algorithms provides no gradient when two bounding boxes have no initial overlap. To overcome this issue, we adopt d-IoU [47] as the distance metric function for AdvTraj in implementing the white-box attack:

$$\mathcal{L}_{DIoU} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}_1, \mathbf{b}_2)}{c^2}, \tag{5}$$

where $\mathbf{b}_1, \mathbf{b}_2$ denote the central points of the two bounding boxes, $\rho(\cdot)$ is the Euclidean distance, and c is the diagonal length of the smallest enclosing box covering the two boxes. Note that the d-IoU distance is always lower bounded by IoU distance $(1 - \text{IoU})$ and provides a non-zero gradient when two bounding boxes have no overlap yet behave similarly to IoU when overlapped. The evaluation of attack success rates still uses the original implementation of the MOT algorithms with default settings.

A.3. Adversarial Trajectory Visualization and Universal Adversarial Maneuver Derivation

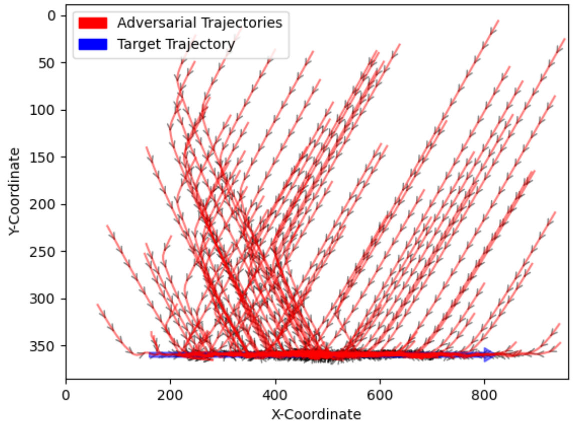
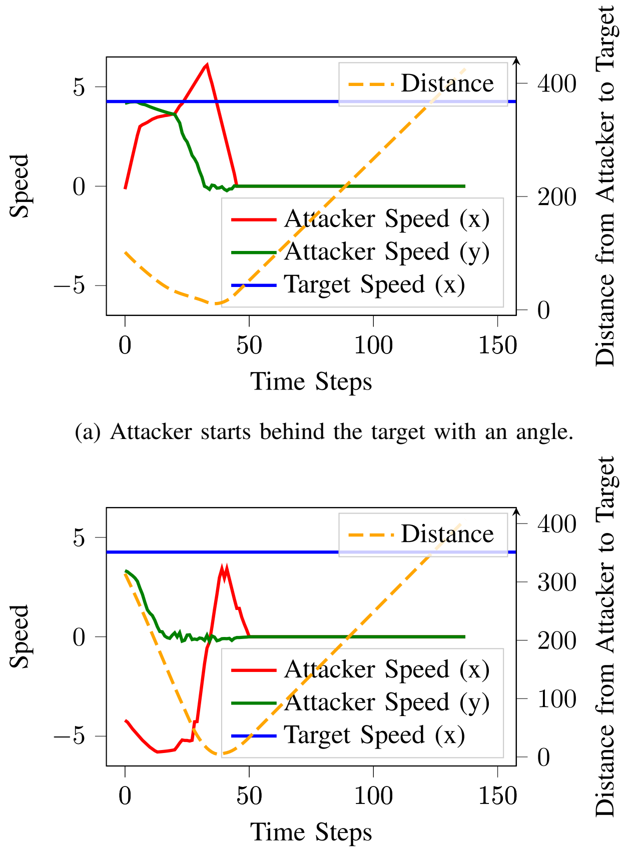


Figure 8: Visualization of generated adversarial trajectories (series of bounding box centers) where the attacker is randomly spawned relative to the target.



(a) Attacker starts behind the target with an angle.

(b) Attacker starts in front of the target with an angle.

Figure 9: Two exemplar adversarial trajectories generated. Note that the acceleration of the attacker when being close the target is in the same direction as the target.

Although AdvTraj is designed to work end-to-end that optimizes for the attacker's 3D physical movement, we lift the 3D-to-2D projection to generate the adversarial trajectories in 2D plane for inspecting the patterns that are agnostic to specific 3D topologies. To constrain the search space and ensure practicality in real-world scenarios, we fix the target's movement along a straight line at a constant speed and randomly initialize the attacker's location relative to the target. We represent the trajectory using the series of bounding box centers and assume that the bounding box shape stays constant. This limits the adversarial effect to being solely dependent on the trajectory itself. The trajectory can be applied in the physical world when the camera is not parallel to the scene since the change in bounding box size (dependent on how close the object is to the image sensor) would be regarded as another 'speed' dimension captured by the KF, where the same effect can be induced by moving along the adversarial trajectories. We plot the generated adversarial trajectories in Figure 8.

We can observe that regardless of the starting position of the attacker relative to the target, the generated trajectories first converge either to the history or projected future path of the target (corresponding to the attacker starting behind/ahead of the target), then maneuver along the same direction as the target with varying speed (non-linear movement). To better understand the movement pattern for each of these two cases, we randomly select an example from each case and decompose the target's velocity in both the x and y axes for visualization as in Figure 9.

In Figure 9a, the attacker starts at a position behind the target. The speed on the y-axis gradually decreases to and stays around zero after the attacker reaches the history path of the target. At the same time, the attacker's speed on the x-axis increases, followed by a decrease to zero (flags the success of ID-Transfer). This motivates the design of the black-box Go-and-Stop tactic for situations where the attacker starts behind the target (even with an angle, the attacker can start by moving to the history path of the target first).

In Figure 9b, the attacker starts at a position ahead of the target. The speed on the y-axis gradually decreases to and stays around zero after the attacker reaches the projected future path of the target. The attacker's speed on the x-axis was initially negative (indicating the attacker is closing the distance to the target by moving in the opposite direction in the x direction). After the attacker is around the target, its speed decreases to zero (from the negative direction of x) and accelerates in the positive direction of x, followed by another decrease to zero (flags the success of ID-Transfer). However, the drastic change in moving direction may be hard to perform in the real world due to the limited maneuverability of the attacker, we design the black-box Stop-and-Go tactic for situations where the attacker starts ahead of the target (even with an angle, the attacker can start by moving to the projected history path of the target and wait for it to pass).

A.4. Results on Vehicle Surveillance

TABLE 4: AdvTraj white-box attack success rates on SORT, transfer attack success rates on other MOT algorithms, and black-box attack success rates, in vehicle surveillance.

	CARLA Simulation	**Real-World Experiments**
SORT	83%	33.3%
ByteTrack	74%	26.6%
OC-SORT	80%	66.6%
Deep OC-SORT	83%	40%
BoT-SORT	56%	26.6%
Strong SORT	40%	0%

For the vehicle surveillance application, the attacker and the target are two vehicles driving in the same direction in two adjacent lanes, where the camera is mounted on the roadside and monitors the roadways horizontally. The target vehicle travels forward at between 25-30 km/h. We simulate the (T1) adversary using CARLA and perform the attack for 100 simulations with random initial positions and target driving speed in Town05. For the (T2) adversary, we mounted an RGB sensor on a tripod in an empty open parking lot monitoring two BMW X5 SUVs (black and white) that were parallel to each other and carried out the attack 15 times. Other simulation/experiment setups are the same as in Section 5.

A.5. Impact of Different Appearances

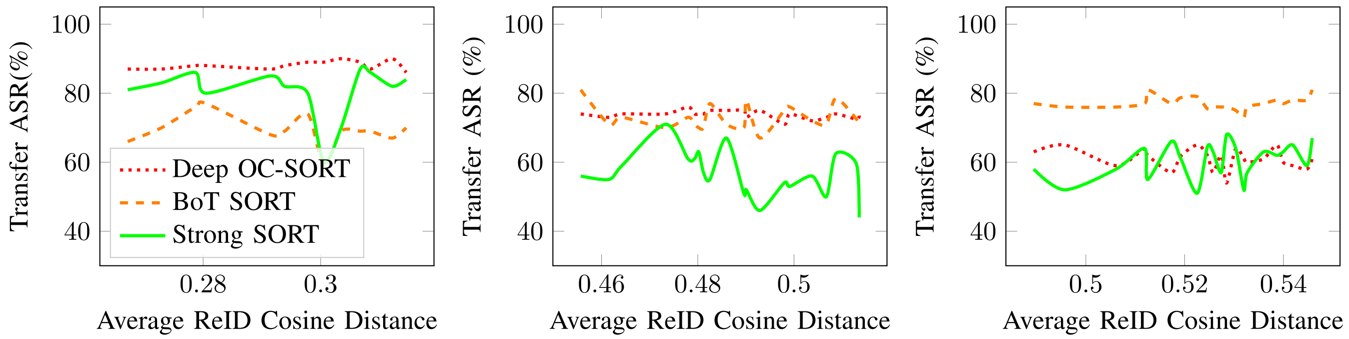
Figure 10 illustrates the impact of different appearances (e.g., color, outfit styles) between attacker and target walkers on the black-box ASRs against MOT algorithms with the pedestrian ReID model enabled. For the same set of generated adversarial trajectories, we mutate the outfits of the attacker and target and quantify their appearance difference by calculating the average cosine distance between the ReID feature vectors across all frames for each CARLA blueprint combination.

The white-box attack conducted in the CARLA simulator demonstrates the effectiveness and transferability of AdvTraj against various MOT algorithms using adversarial trajectories. However, due to the limited availability of walker actor blueprints offered by CARLA, the assessment of the impact of different pedestrian appearances on transfer attacks cannot extend to a wider range beyond the 25 combinations of outfits. Nevertheless, the different blueprint combinations we evaluated in simulation and the distinct appearance of attacker and target (black/white) in real-world experiments still indicate that the impact of the adversarial trajectories/maneuvers remain potent, even if the two tracked objects appear visually distinct.

A.6. End-to-End Impact on Autonomous Driving

Case Study Setup. As shown in Figure 11, we construct the AD-parallel scenario in CARLA Town01 where two pedestrians on the sidewalk walk parallel to the AD system, controlled by the Baidu Apollo agent through a CARLA-Apollo bridge [59]. The AD system starts 200 meters ahead of the starting positions of the two pedestrians and drives autonomously under a routing request to reach a destination behind them. We visualize the trajectory prediction results of the AD on the two pedestrians and evaluate the corresponding driving decisions in both benign and adversarial cases. In the benign case, the two pedestrians walk parallel to the AD system at a constant speed of 1 m/s. In the benign case where no attack occurs, the AD system is expected to pass the two pedestrians safely, as the two pedestrians on the sidewalk should not be expected to enter the roadway. In the adversarial case, the attacker walker starts at a position behind the target and performs the black-box Go-and-Stop tactic, and its ID is transferred to the target as perceived by the AD system. The mismatch between detections and trajectory histories could lead to errors in predicting future pedestrian trajectories, which could cause unpredictable and potentially dangerous driving decisions.

Impact of ID-Transfer. As shown in Figure 11a, before the attack, the AD system produces trajectory predictions for the

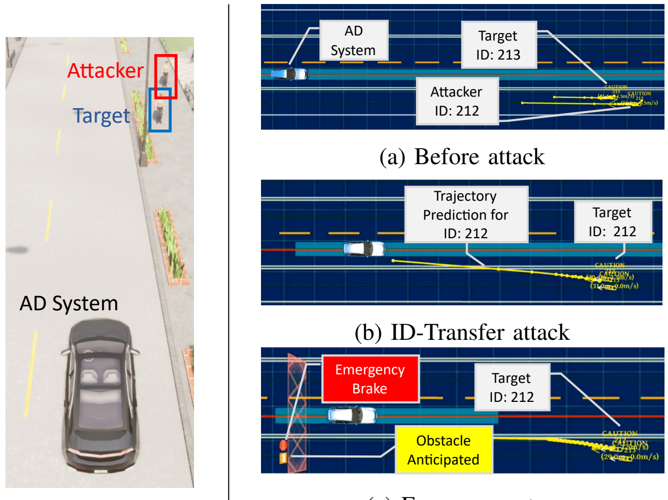


(a) Pedestrian Surveillance

(b) AD-Perpendicular

(c) AD-Parallel

Figure 10: Transfer ASRs with different attacker-target appearance combinations for three applications.



(a) Before attack

(b) ID-Transfer attack

(c) Emergency stop

Figure 11: Consequence of ID-Transfer on Baidu Apollo AD system: (left) Scenario setup. (a) Before the attack. (b) The attacker walker stops walking once its ID is transferred. The history trajectory of the attacker is wrongly appended to the target's trajectory based on the assigned ID, resulting in an "upward momentum" that confuses the prediction module, where the predicted trajectory of the target intersects with the vehicle's roadway. (c) The AD system performs an emergency brake to avoid a wrongly anticipated collision.

two pedestrians consistent with their true movement along straight paths at a constant speed. The same result is observed for the benign case where the AD runs properly and passes pedestrians without interruption. However, after the attacker performs the ID-Transfer attack, its ID is assigned to the target object, resulting in an inconsistent history update and a trajectory that "pivots" towards the driveway as perceived by the AD. This erroneous trajectory causes the AD system to anticipate that the target crosses the driveway as shown in Figure 11b. To avoid this anticipated crash, the AD decides to perform an emergency brake as seen in Figure 11c, which leads not only to an uncomfortable experience for the passenger but also potentially to a rear-end collision by the following vehicle. Notably, the attack appears stealthy to humans, since both the attacker and target walkers are only walking along straight lines with varying speeds without explicit signs to disrupt the vehicle's operation.

A.7. Relation to ID-Switch

AdvTraj aims at transferring the ID assignments between the attacker and a targeted object, which is more stealthy than the general ID-Switch attack, where preservation of original IDs is not guaranteed. Yet, even when an attacker does not successfully transfer its ID to the target using AdvTraj, we observe that the attempt usually results in an ID-Switch for the attacker. For example, in the black-box attack experiments for surveillance by (T2) adversary, BoT-SORT and StrongSORT have 5% and 0% ID-Transfer rates between the attacker and the target, but have 80% and 60% ID-Switch rates for the attacker, respectively. This is because the highly non-linear adversarial movement voids the linear motion assumption made by the MOT algorithms, leading to a large distance between the tracker's linearly predicted location and the ground truth location. By employing adversarial trajectories alone, ID-Switch (the evaluated attacker goal of previous works [11–14] by attacking OD) can be achieved without fooling the OD model of MOT.

Note that the adversarial loss function of AdvTraj encourages swapping the IDs of the attacker and target, where both the attacker's and target's IDs are preserved yet wrongly assigned. In the case of (T1) adversary and assuming accurate bounding box detections, swapped IDs imply successful ID-Transfer. In real-world execution, natural tracker loss may occur due to occlusions and/or detection errors (missing detections) if the target is occluded for a number of frames exceeding the threshold set by the MOT algorithm. Nevertheless, for the attacker intended to escape surveillance without raising suspicions or cause incorrect trajectory predictions, it suffices to achieve ID-Transfer where its own ID is preserved but transferred to the target.