BankTweak: Adversarial Attack Against Multi-Object Trackers by Manipulating Feature Banks

Woojin Shin[1], **Donghwa Kang**[2], **Daejin Choi**[3], **Brent Byunghoon Kang**[2], **Jinkyu Lee**[4] and **Hyeongboo Baek**[1]
[1]University of Seoul, Seoul, Korea
[2]Korea Advanced Institute of Science and Technology, Daejeon, Korea
[3]Ewha Womans University, Seoul, Korea
[4]Sungkyunkwan University, Suwon, Korea
jinkyu.lee@skku.edu, hbbaek@uos.ac.kr

Abstract

Modern multi-object tracking (MOT) predominantly relies on the tracking-by-detection paradigm to construct object trajectories. Traditional MOT attacks primarily degrade detection quality in specific frames only, lacking *efficiency*, while state-of-the-art (SOTA) approaches induce persistent identity (ID) switches by manipulating object positions during the association phase, even after the attack ends. In this paper, we reveal that these SOTA attacks can be easily counteracted by adjusting distance-related parameters in the association phase, exposing their lack of *robustness*. To overcome these limitations, we propose BankTweak, a novel adversarial attack targeting feature-based MOT systems to induce persistent ID switches (*efficiency*) without modifying object positions (*robustness*). BankTweak exploits a critical vulnerability in the Hungarian matching algorithm of MOT systems by strategically injecting altered features into feature banks during the association phase. Extensive experiments on MOT17 and MOT20 datasets, combining various detectors, feature extractors, and trackers, demonstrate that BankTweak significantly outperforms SOTA attacks up to 11.8 times, exposing fundamental vulnerabilities in the tracking-by-detection framework.

1 Introduction

Multi-object tracking (MOT) is a fundamental perception task aimed at constructing motion trajectories of objects across consecutive frames. Modern DNN-based *tracking-by-detection* frameworks consist of two stages: detecting objects of interest in each frame (*detection*) and associating these detections with existing trajectories (*association*) [Wojke *et al.*, 2017]. In the association phase, a CNN-based model (e.g., OSNet [Zhou *et al.*, 2019]) extracts object features, which are stored in a *feature bank* if matched using feature-based similarity or motion-based IoU (Intersection over Union). Feature-based matching initially pairs objects with the highest feature similarity, followed by motion-based IoU matching for unmatched objects.

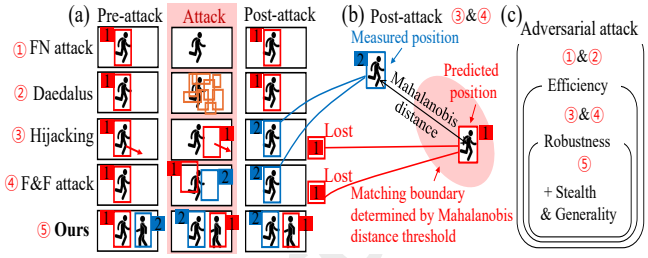


Figure 1: (a) Comparison between existing adversarial attacks ①–④ and BankTweak ⑤, (b) the principles behind ID switch induction in Hijacking ③ and the F&F attack ④, and (c) the diagram presenting the features of adversarial attacks.

Despite the widespread application and significance of MOT systems, studies on adversarial attacks and defense mechanisms remain limited, falling into two categories: i) targeting the detection phase to generate false negatives and false alarms, and ii) state-of-the-art (SOTA) attacks disrupting the association phase to cause identity (ID) switches by manipulating object positions. Category i) attacks, such as the false negative (FN) attack [Lu *et al.*, 2017] (① in Fig. 1) and Daedalus [Wang *et al.*, 2021] (②), degrade detection performance in attacked frames by generating false negatives and false alarms, respectively. However, as these methods focus solely on the detection phase, their effects are limited to the targeted frames, lacking *efficiency*. For instance, in FN attack and Daedalus scenarios (Fig. 1(a)), the object with ID 1 retains the same ID in post-attack frames, resulting in no impact on post-attack accuracy.

In category ii), Hijacking [Jia *et al.*, 2020] (③) disrupts the Kalman filter [Kalman, 1960] by manipulating detection boxes away from their correct velocity, potentially triggering ID switches. The F&F attack [Zhou *et al.*, 2023] (④) removes the target object and surrounds it with false alarms, resulting in persistent ID switches even after the attack concludes. As shown in Fig.1(b), these attacks increase the Mahalanobis distance [De Maesschalck *et al.*, 2000] between the measured and predicted positions in the post-attack frame. This distance

exceeds the matching boundary, leading to misidentification and the assignment of a new (false) ID (e.g., ID 2). A key misunderstanding of such SOTA attacks is the assumption that persistent ID switches can only be caused by shifting object positions beyond the matching boundary.

In this paper, exploiting this misunderstanding, we first reveal that increasing the Mahalanobis distance threshold (expanding the matching boundary) in prediction models, such as the Kalman filter, effectively counteracts category ii) attacks with minimal accuracy loss (as detailed in Sec. 4), thereby exposing their inherent lack of *robustness*. Then, overcoming the inherent limitations in i) and ii), we propose BankTweak, a novel adversarial attack that establishes a new paradigm for MOT attacks by inducing persistent ID switches (*efficiency*) while preserving object positions (*robustness*). This is achieved through a two-step process: (i) injecting targeted features into the feature bank of selected object pairs without causing ID switches (Step 1: groundwork), and (ii) exploiting this setup to induce permanent ID switches (Step 2: ID Switch). As described in Sec. 3 and illustrated in Figs. 3(b) and (c), inducing persistent ID switches independently of object positions is inherently challenging due to the properties of feature banks. However, BankTweak overcomes this limitation by exploiting vulnerabilities in the Hungarian matching algorithm used by feature-based MOT systems. Notably, BankTweak is capable of simultaneously attacking multiple object pairs within a single frame

As an additional key advantage, unlike the F&F attack, which increases the risk of detection by defenders by generating numerous false alarm objects, BankTweak completely avoids producing false alarms, thereby significantly enhancing its *stealth*. Moreover, unlike the Hijacking attack requiring model-specific design like Kalman filter information, BankTweak does not rely on such specifics, ensuring greater *generality*. Fig. 1(c) represents the characteristics of MOT attacks, including BankTweak. Importantly, BankTweak is not tailored to a specific tracker, making it compatible with *most* DNN-based tracking-by-detection methods that use a feature bank and a two-stage association approach (i.e., feature-based and subsequent IoU-based).

To demonstrate the applicability, BankTweak is applied to three multi-object trackers (DeepSORT [Wojke *et al.*, 2017], StrongSORT [Du *et al.*, 2023], and MOTDT [Chen *et al.*, 2018]) using a diverse set of detectors: one-stage (YOLOX [Ge *et al.*, 2021]), two-stage (Faster R-CNN [Ren *et al.*, 2015]), anchor-free (FoveaBox [Kong *et al.*, 2020]), and transformer-based (DETR [Zhu *et al.*, 2021]), along with four feature extractors (OSNet [Zhou *et al.*, 2019], ResNet [He *et al.*, 2016], MobileNet [Howard, 2017], and MLFN [Chang *et al.*, 2018]). Comprehensive experiments conducted on the MOT17 [Milan *et al.*, 2016] and MOT20 [Dendorfer *et al.*, 2020] datasets demonstrate that our approach significantly outperforms SOTA attacks up to 11.8 times, revealing the vulnerability of the tracking-by-detection framework to BankTweak.

Our contributions are as follows.

- We reveal that increasing the Mahalanobis distance threshold in MOT systems, effectively counteracts SOTA attacks with minimal accuracy loss.
- We propose BankTweak, a novel adversarial attack characterized by efficiency and robustness, with benefits of practicality and generality.
- We demonstrate BankTweak's applicability by deploying it to multiple combinations of MOT trackers with detectors, achieving superior performance over existing attacks in extensive public dataset experiments.

2 Related Work

Multi-object tracking (MOT) aims to construct object motion trajectories across frame sequences [Luo *et al.*, 2021]. Most MOT methods utilize the tracking-by-detection paradigm, featuring online (real-time) and offline (post-refinement) approaches [Zhang *et al.*, 2022; Zhou *et al.*, 2020; Dai *et al.*, 2021]. Detectors identify objects, while trackers associate them using motion (e.g., Kalman filter [Kalman, 1960]) and appearance cues [Zhang *et al.*, 2021]. Transformer-based methods [Meinhardt *et al.*, 2022; Sun *et al.*, 2020] now integrate detection and association.

Adversarial attacks are studied in detection, tracking, and segmentation [Wang *et al.*, 2021; Jia *et al.*, 2020; Xie *et al.*, 2017], with some real-world demonstrations [Xu *et al.*, 2020]. Attacks on single-object tracking often target accuracy, and detection attacks cause missed/false detections. However, MOT's complexity poses challenges for these methods. Specific MOT attacks include targeting the Kalman filter [Jia *et al.*, 2020] or inducing ID switches via detection failures, as with the F&F attack [Zhou *et al.*, 2023].

3 Method

3.1 Attack Formulation

We consider tracking-by-detection MOT systems, consisting of detection and association phases, discussed in Sec. 1. BankTweak operates under the assumption of a white box attack, with the attacker knowing the detector and feature extractor models for the iterative execution of detection and feature extraction across attack frames. BankTweak only needs five frames for an attack without any false alarms (addressing *stealth*), and does not necessitate motion prediction of objects (addressing *generality*). A common attack scenario of BankTweak can be a man-in-the-middle attack, where input packets are intercepted via DNS spoofing, perturbed, and then re-injected into a server-based CCTV system.

Consider an input video comprised of N sequential RGB frames $I \in \mathbb{R}^{W\times H\times 3}$, represented as $\mathbb{V} = \{I_1, I_2, \cdots, I_N\}$. We target a sequence of five consecutive frames starting from the $(t{+}1)$-th frame (for $1{\leq}t$) for our attack, denoted by $\mathbb{S} = \{I_{t+1}, I_{t+2}, I_{t+3}, I_{t+4}, I_{t+5}\}$. Let $\tilde{I}_t$ be a frame created by adding a perturbation δ to I_t. By incorporating perturbations into every frame in $\mathbb{S}$, we generate $\tilde{\mathbb{S}} = \{\tilde{I}_{t+1}, \tilde{I}_{t+2}, \tilde{I}_{t+3}, \tilde{I}_{t+4}, \tilde{I}_{t+5}\}$ (for $t{+}5{<}N$). Substituting $\mathbb{S}$ in $\mathbb{V}$ with $\tilde{\mathbb{S}}$ yields $\tilde{\mathbb{V}} = \{I_1, I_2, \cdots, I_t, \tilde{I}_{t+1}, \tilde{I}_{t+2}, \cdots, \tilde{I}_{t+5}, I_{t+6}, I_{t+7} \cdots, I_N\}$.

For the given target input frame I, the detector $D(\cdot|\theta_D)$ parameterized by θ_D, feature extractor $E(\cdot,\cdot|\theta_E)$ parameterized

Algorithm 1 BankTweak attack

Input: target frame sequence $\mathbb{S}$, object detector $D(\cdot)$, feature extractor $E(\cdot,\cdot)$ **Output:** perturbed frame sequence $\tilde{\mathbb{S}}$

1: $\tilde{\mathbb{S}} = [\,]$
2: **for** I from I_{t+1} to I_{t+5} in $\mathbb{S}$ **do**
3: $\quad\mathbb{F}^* = E(D(I), I)$ /* get targeted feature set and loss function (Sec. 3.2) */
4: $\quad\mathbb{F}, \mathcal{L} \leftarrow$ get_targeted_features($\mathbb{F}^*$) /* solve perturbation with Eqs. (1) and (2) */
5: $\quad\tilde{I} \leftarrow$ solve_perturbation($\mathbb{F}, \mathcal{L}, I, D(\cdot), E(\cdot,\cdot)$)
6: $\quad\tilde{\mathbb{S}}$.append($\tilde{I}$)
7: **end for**
8: **return** $\tilde{\mathbb{S}}$

by θ_E, and target features $\mathbb{F}$, BankTweak finds perturbation δ formulated by

$$\delta = \underset{\delta, ||\delta||_\infty < \epsilon}{\arg\min}\ \mathcal{L}(E(D(I+\delta|\theta_D), I+\delta|\theta_E), \mathbb{F}), \quad (1)$$

where $D(\cdot|\theta_D)$ processes an input frame I to identify the detected object set $\mathbb{O}$ and $E(\cdot,\cdot|\theta_E)$ extracts feature set $\mathbb{F}^*$ from $\mathbb{O}$. BankTweak uses PGD to iteratively derive a perturbation δ that minimizes $\mathcal{L}$ under an ℓ_∞-norm constrain ($\mathcal{L}$ will be further detailed in Sec. 3.3), which is

$$\delta^{r+1} = \text{clip}_{[-\epsilon,\epsilon]\cap[-I,1-I]} (\delta^r + \alpha \text{sgn}(\nabla_\delta \mathcal{L}(E(D(I+\delta|\theta_D), I+\delta|\theta_E), \mathbb{F}))), \quad (2)$$

where α and ϵ represent the amount of change per pixel and the maximum change allowed, respectively, while ∇ and $\text{sgn}(\cdot)$ are functions for performing the gradient operation and extracting the sign of the gradient, respectively. BankTweak initializes the first perturbation δ to zero and iterates R times to compute the final δ. During these iterations, δ must adhere to an ℓ_∞-norm constraint, ensuring the perturbed frame $\tilde{I}$ remains within the [0, 1] range.

Alg. 1 outlines the attack process of BankTweak. It takes as input the target frame sequence $\mathbb{S}$, object detector $D(\cdot)$, and feature extractor $E(\cdot,\cdot)$, producing the perturbed frame sequence $\tilde{\mathbb{S}}$ as output. For each input frame I, BankTweak performs the detection to obtain the object set $\mathbb{O}$ and then conducts feature extraction based on $\mathbb{O}$ to extract the feature set $\mathbb{F}^*$ (Line 3). Subsequently, it determines the designated $\mathbb{F}$ and $\mathcal{L}$ for each attack frame based on $\mathbb{F}^*$ (Line 4). Utilizing the derived $\mathbb{F}$, $\mathcal{L}$, and the models $D(\cdot)$ and $E(\cdot,\cdot)$, it computes the perturbed frame $\tilde{I}$ using Eqs. (1) and (2) (Line 5), which is then added to $\tilde{\mathbb{S}}$. The detector $D(\cdot)$ is used for cropping the detected object from the input image after performing detection, and the perturbation is determined through the model $E(\cdot,\cdot)$ (Line 5). This procedure is repeated for the length of the input frame sequence $\mathbb{S}$, which is five (i.e., from I_{t+1} to I_{t+5}), and ultimately returns the perturbed frame sequence $\tilde{\mathbb{S}}$ (Line 8).

3.2 BankTweak Mechanism

The primary objective of BankTweak is to switch the IDs of target objects Ⓐ and Ⓑ, ensuring these changes remain constant, even after the completion of the attack. This process unfolds in two steps. (i) Initially, BankTweak systematically injects perturbed features into the feature banks of Ⓐ and Ⓑ, without initiating an ID switch. This preparatory action lays the groundwork for the next step. (ii) Subsequently, leveraging the altered feature banks established in (i), BankTweak executes the ID switch for Ⓐ and Ⓑ, effectively achieving the intended consistent ID switch even after the attack. Fig. 2 presents the overall process by which BankTweak induces an ID switch between a pair of objects Ⓐ and Ⓑ. For a clean frame I_t, features A and B are extracted from objects Ⓐ and Ⓑ, respectively, and are assigned ID 1 and ID 2. These features are then stored in the feature banks of their corresponding objects. Consider $A|B$ as feature A generated from B through an adversarial example mechanism (e.g., PGD [Madry *et al.*, 2017]), which appears as B to humans but is identified as A by the deployed model. BankTweak selects the object pair Ⓐ and Ⓑ for an ID switch in each frame I_t, fundamentally choosing Ⓐ and Ⓑ randomly without awareness of the Mahalanobis distance threshold, thereby satisfying generality.

Step 1: Groundwork. It performs the following for the first three attacked frames:

$\tilde{I}_{t+1}$: Define X and Y as the dummy features that exhibit a significantly large cosine distance from A and B, ensuring they are distinctly different. By definition, $X|A$ and $Y|B$ have a high cosine distance (e.g., 0.9) from A and B, respectively, and are injected into the feature banks of Ⓐ and Ⓑ through IoU matching; it is assumed that features are only considered for feature-based matching when they have a cosine distance of 0.5 or less (called cosine distance threshold).

$\tilde{I}_{t+2}$: It places $B|A$ into Ⓐ's feature bank and $Y|B|B$ into Ⓑ's feature bank, leveraging the very low cosine distance between $Y|B|B$ and $Y|B$ (e.g., 0.02 in Fig. 2) once it is inserted into Ⓑ's bank in $\tilde{I}_{t+1}$. Being generated from A, $B|A$ exhibits a relatively low cosine distance (e.g., 0.07 in Fig. 2).

$\tilde{I}_{t+3}$: It places $A|B$ into Ⓑ's feature bank and $X|A|A$ into Ⓐ's feature bank using the similar property in $\tilde{I}_{t+2}$.

Fig. 3(a) presents our experimental result for two distinct scenarios, each featuring varying converging cosine distances when a source feature is subjected to up to 150 perturbations to derive a specific target feature, as outlined in Eq. (4). For instance, when producing $Y|B|B|A$ (in $\tilde{I}_{t+4}$), the source feature is A, targeting the feature $Y|B|B$ (in $\tilde{I}_{t+2}$). Conversely, $Y|B|B$ is derived from B, resulting in a cosine distance of 0.07 between $Y|B|B|A$ and $Y|B|B$ due to the disparity in their source features. On the other hand, for the production of $Y|B|B$ (in $\tilde{I}_{t+2}$), B acts as the source feature with $Y|B$ (in $\tilde{I}_{t+1}$) as the target. Here, $Y|B$, created from the same source B, leads to a minimal cosine distance of 0.02 between $Y|B|B$ and $Y|B$. It might seem straightforward to induce an

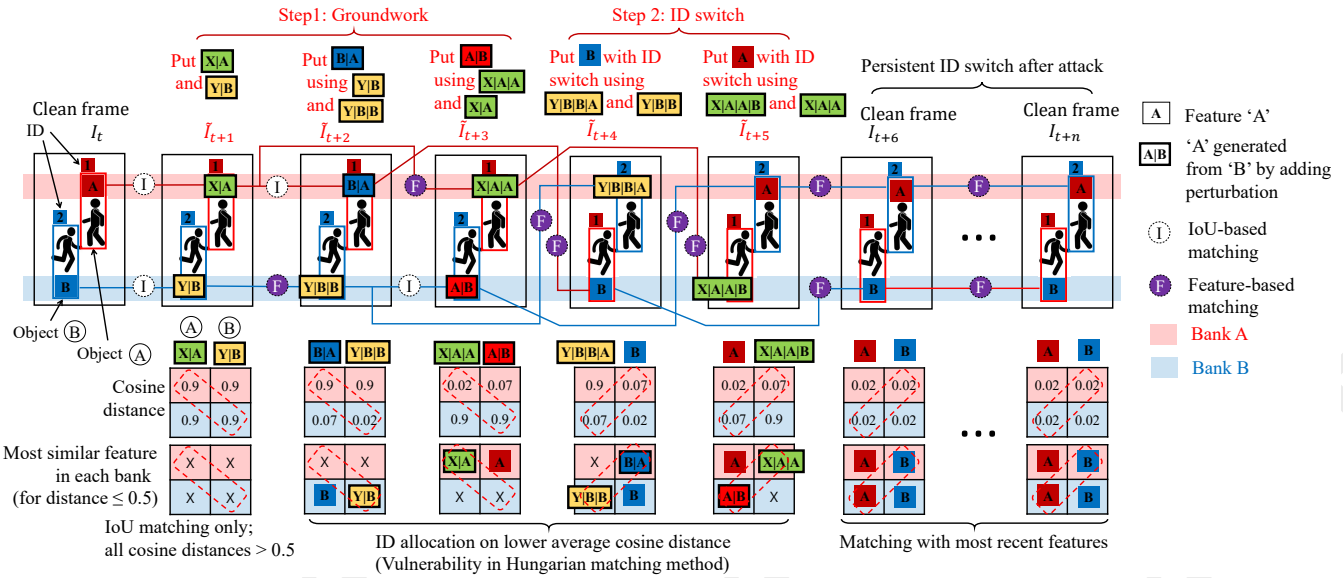


Figure 2: Overall process by which BankTweak induces a persistent ID switch between a pair of objects Ⓐ and Ⓑ, even after the attack ends, across five frames from $\tilde{I}_{t+1}$ to $\tilde{I}_{t+5}$. In step 1, Desired generated features are first injected into the feature banks of Ⓐ and Ⓑ in $\tilde{I}_{t+1}$–$\tilde{I}_{t+3}$. In step 2, a ID switch incurs in $\tilde{I}_{t+4}$–$\tilde{I}_{t+5}$ by exploiting a vulnerability in the Hungarian matching method, where ID allocation for objects relies on matching across frames with lower average cosine distance. The persistent ID switch continues to occur in the post-attack frames, $\tilde{I}_{t+6}$ through $\tilde{I}_{t+n}$, where objects Ⓐ and Ⓑ remain present.

ID switch by injecting $B|A$ and $A|B$ into the feature banks of Ⓐ and Ⓑ in Step 1. However, Fig. 3(b) illustrates that because the feature banks of Ⓐ and Ⓑ already include A and B, any ID switch in $\tilde{I}_{t+1}$ reverts to the original IDs by $\tilde{I}_{t+2}$ post-attack.

Step 2: ID Switch. This step involves the following for the next two frames:

$\tilde{I}_{t+4}$: It places $Y|B|B|A$ into Ⓐ's feature bank without creating any perturbation for Ⓑ. B in $\tilde{I}_{t+4}$ demonstrates a significantly low cosine distance (i.e., 0.02) to B in I_t, whereas $Y|B|B|A$ exhibits a high cosine distance (i.e., 0.9) with the features in Ⓐ's bank. Given the cosine distances of 0.07 between $Y|B|B|A$ and $Y|B|B$, and 0.07 between B and $B|A$, the Hungarian algorithm [Kuhn, 1955] allocates IDs based on the lower average cosine distance, prompting an ID switch.

$\tilde{I}_{t+5}$: It places $X|A|A|B$ into Ⓑ's feature bank, and similarly, no perturbation is created for Ⓐ. Similar reasoning to $\tilde{I}_{t+4}$ causes an ID switch for $\tilde{I}_{t+5}$.

As a result, from I_{t+6} onward, A in Ⓐ matches with A having ID 2 in $\tilde{I}_{t+5}$ (rather than A with ID 1 in I_t), and B in Ⓑ matches with B having ID 1 in $\tilde{I}_{t+4}$ (rather than B with ID 2 in I_t), thus continuous ID switches occur without further attacks. One might assume that directly injecting A and B into the feature banks of Ⓐ and Ⓑ for $\tilde{I}_{t+4}$ in Step 2 is a straightforward approach to trigger an ID switch. However, as shown in Fig. 3(c), such an action does not lead to an ID switch due to the pre-existing A and B in the respective feature banks of Ⓐ and Ⓑ. To this end, BankTweak employs a meticulous strategy that exploits the vulnerability (i.e., ID allocation on lower average cosine distance) of the Hungarian algorithm in Step 2, based on the groundwork conducted in Step 1.

3.3 Solving Perturbations

BankTweak employs cosine distance to evaluate the similarity between the features of two objects, facilitating the creation of the target feature set $\mathbb{F}$ from an initial feature set $\mathbb{F}^*$, which is derived by

$$\mathcal{C}(A, B) = 1 - \frac{A \cdot B}{|A||B|}, \tag{3}$$

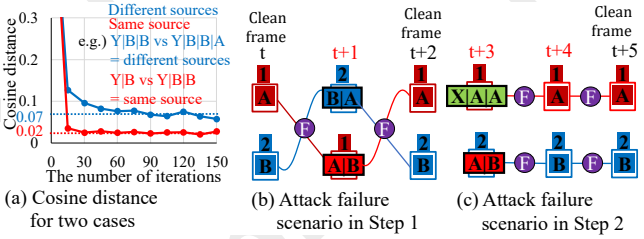
where A and B are feature vectors of two distinct objects, each in $\mathbb{R}^{1\times 512}$. Eq. (3) produces values within the [0, 2] range, with lower values denoting higher similarity and higher values indicating greater dissimilarity between the features of two objects.

For a frame I, which includes multiple objects, we define the extracted feature set as $\mathbb{F}^*$ and its target feature set as $\mathbb{F}$. For each feature $F_i^* \in \mathbb{F}^*$, $F_i \in \mathbb{F}$ represents its corresponding target feature. BankTweak computes the loss for each feature set $\mathbb{F}^*$, aggregates these losses, and applies the perturbation collectively. This process employs a specific loss function formulated as

$$\mathcal{L}^s(\mathbb{F}^*, \mathbb{F}) = \sum_{F_i^* \in \mathbb{F}^*, F_i \in \mathbb{F}} \mathcal{C}(F_i^*, F_i), \tag{4}$$

$$\mathcal{L}^d(\mathbb{F}^*, \mathbb{F}) = -\sum_{F_i^* \in \mathbb{F}^*, F_i \in \mathbb{F}} \mathcal{C}(F_i^*, F_i). \tag{5}$$

For each feature $F_i^* \in \mathbb{F}^*$ and its target feature $F_i \in \mathbb{F}$, the loss $\mathcal{L}^s$ signifies that a lower value increases the similarity between F_i^* and F_i. Conversely, a higher value of $\mathcal{L}^d$



(a) Cosine distance for two cases (b) Attack failure scenario in Step 1 (c) Attack failure scenario in Step 2

Figure 3: (a) Comparison of cosine distance for two cases involving the same and different object sources, and (b)–(c) two attack failure scenarios of BankTweak.

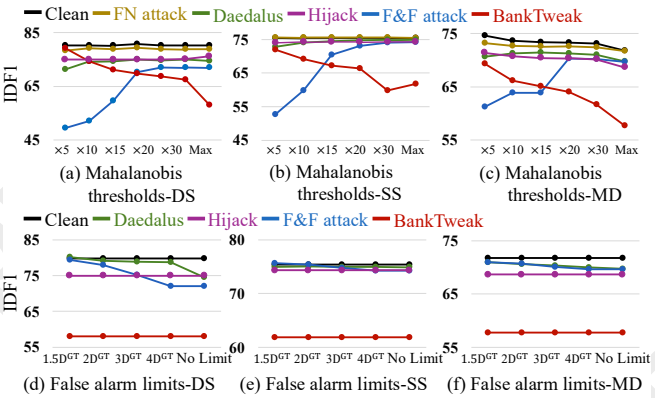


Figure 4: IDF1 scores across (a)–(c) varying Mahalanobis distance thresholds and (d)–(f) false alarm limits for three distinct trackers.

decreases the similarity between F_i^* and F_i. For instance, the goal for $\tilde{I}_{t+1}$ is to generate $X|A$ and $Y|B$, thus the target feature set $\mathbb{F} = \{F_1 = A, F_2 = B\}$, and $X|A$ and $Y|B$ are created to have the maximum possible cosine distance from A and B, respectively, using Eq. (5). On the other hand, for $\tilde{I}_{t+2}$, aiming to generate $B|A$ and $Y|B|B$, the target feature set $\mathbb{F} = \{F_1 = B, F_2 = Y|B\}$, and $B|A$ and $Y|B|B$ are produced to be as close as possible to A and B, respectively, using Eq. (4). The feature sets $\mathbb{F}^*$, target feature set $\mathbb{F}$, and the loss function for each attack frame are determined as follows.

$\tilde{I}_{t+1}$: $\mathbb{F}^* = \{F_1^* = A, F_2^* = B\}$ and $\mathbb{F} = \{F_1 = A, F_2 = B\}$ with $\mathcal{L}^d(\mathbb{F}^*, \mathbb{F})$,
$\tilde{I}_{t+2}$: $\mathbb{F}^* = \{F_1^* = A, F_2^* = B\}$ and $\mathbb{F} = \{F_1 = B, F_2 = Y|B\}$ with $\mathcal{L}^s(\mathbb{F}^*, \mathbb{F})$,
$\tilde{I}_{t+3}$: $\mathbb{F}^* = \{F_1^* = A, F_2^* = B\}$ and $\mathbb{F} = \{F_1 = X|A, F_2 = A\}$ with $\mathcal{L}^s(\mathbb{F}^*, \mathbb{F})$,
$\tilde{I}_{t+4}$: $\mathbb{F}^* = \{F_1^* = A\}$ and $\mathbb{F} = \{F_1 = Y|B|B\}$ with $\mathcal{L}^s(\mathbb{F}^*, \mathbb{F})$, and
$\tilde{I}_{t+5}$: $\mathbb{F}^* = \{F_1^* = B\}$ and $\mathbb{F} = \{F_1 = X|A|A\}$ with $\mathcal{L}^s(\mathbb{F}^*, \mathbb{F})$.

4 Evaluation

4.1 Experiment Setting

Metrics. We compare the performance of the considered approaches regarding efficiency, robustness, and stealth. BankTweak inherently ensures generality as it does not require any information about the prediction model. For efficiency, we use standard MOT accuracy metrics such as IDF1 [Ristani *et al.*, 2016] and HOTA [Luiten *et al.*, 2021]. IDF1 is defined as IDF1 $= 2 \times \frac{\text{TP}}{2\times\text{TP}+\text{FP}+\text{FN}}$, where TP, FP, and FN are true positives, false positives, and false negatives. HOTA is given by HOTA $= \sqrt{\text{DetA} \cdot \text{AssA}}$, with DetA $= \frac{\text{TP}}{\text{TP}+\text{FN}+\text{FP}}$ and AssA $= 1 - \frac{\text{IDsw}}{\text{GTtrack}}$, where IDsw is the number of ID switches and GTtrack is the total number of tracklets in the ground truth. These metrics exclude attack frames for accuracy. For robustness, we measure accuracy by varying the Mahalanobis distance threshold. For stealth, we measure ρ^{Det} and ρ^{ID} to evaluate the system's ability to reduce new objects (mainly false alarms) during attack frames. ρ^{Det} is the ratio of the average detection increase per attack frame to the ground truth GT_t, while ρ^{ID} quantifies the increase in ID counts.

Dataset. Experiments are conducted using the MOT17 and MOT20 pedestrian tracking datasets. Each dataset is split into two halves: one for training the considered detection model and the other for evaluation. The MOT17 and MOT20 datasets are further divided into 30-frame segments, yielding 83 and 148 segments, respectively. Experiments target each segment's (15–19)-th frames for attacks to accumulate features in the objects' feature banks over five frames, ensuring accurate evaluation of BankTweak's potential effects in practical tracking applications.

Implementation Details. To demonstrate the applicability, BankTweak is applied to three prominent multi-object trackers (DeepSORT, StrongSORT, and MOTDT denoted by DS, SS, and MD, respectively) with one-stage (YOLOX), two-stage (Faster-RCNN), anchor-free (FoveaBox) and transformer (DETR) detectors. We consider OSNet, ResNet, MobileNet, and MLFN as feature extractors. The feature-based matching threshold is $\lambda_{app} = 0.2$, and IoU-based matching threshold is $\lambda_{IoU} = 0.7$. Attack parameters are $\epsilon = 4/255$ and $\alpha = 1/255$. In BankTweak, dissimilarity loss $\mathcal{L}^d$ succeeds when feature similarity exceeds $\lambda_{app} = 0.2$, and similarity loss $\mathcal{L}^s$ requires cosine distance to be less than $\lambda_{app} = 0.2$. Empirically, iterations for $\mathcal{L}^d$ are set to $R^d = 10$, and for $\mathcal{L}^s$, $R^s = 150$.

4.2 Comparison to Existing Methods

We evaluate our approach against four existing attacks: the FN attack, Daedalus, Hijacking, and F&F attack.

Efficiency and Robustness. Tab. 1 shows attack performance under the maximum Mahalanobis distance threshold, counteracting Hijacking and F&F attack with minimal accuracy drop (see Figs. 4(a)–(c)) for YOLOX and OSNet. The experimental results for combinations with other detectors and feature extractors are provided in the supplementary material. Tab. 1 demonstrates that BankTweak significantly decreases the IDF1 and HOTA score across all trackers. For instance, BankTweak reduces the IDF1 score by 13.57 for SS on MOT17, compared to 1.15 for F&F, achieving up to an 11.8-fold (13.57/1.15) improvement. It also significantly increases IDsw (e.g., 1847% for MD on MOT20) while de-

Dataset	Tracker	Attacker	IDF1	HOTA	IDsw	DetA	AssA	FN	FP	ρ^{ID}	ρ^{Det}
MOT17	DS	Clean	79.78	66.73	173	58.35	77.57	0.304	0.022	0.93	0.82
		FN attack	78.90 (-0.88)	66.10 (-0.63)	193 (+11%)	57.95	76.70	0.307	0.025	0.00	0.00
		Daedalus	74.48 (-5.30)	63.28 (-3.45)	382 (+120%)	57.08	71.23	0.311	0.033	7.18	2.01
		Hijacking	75.37 (-4.41)	63.32 (-3.41)	394 (+127%)	57.51	70.84	0.305	0.029	0.93	0.82
		F&F	72.03 (-7.75)	61.60 (-5.13)	493 (+184%)	56.57	68.11	0.313	0.039	3.13	2.59
		Ours	58.01 (-21.77)	51.15 (-15.58)	877 (+406%)	54.30	56.25	0.336	0.057	0.92	0.82
	SS	Clean	75.41	64.38	111	54.28	77.82	0.342	0.096	0.93	0.83
		FN attack	75.43 (+0.02)	64.43 (+0.05)	96 (-13%)	54.21	78.02	0.342	0.096	0.00	0.00
		Daedalus	74.68 (-0.73)	63.86 (-0.52)	136 (+22%)	54.10	76.77	0.342	0.099	7.18	2.01
		Hijacking	74.32 (-1.09)	63.75 (-0.63)	163 (+46%)	54.03	76.68	0.345	0.098	0.93	0.77
		F&F	74.26 (-1.15)	63.43 (-0.95)	158 (+42%)	53.84	76.02	0.344	0.103	3.13	2.59
		Ours	61.84 (-13.57)	54.35 (-10.03)	712 (+541%)	52.94	56.77	0.353	0.109	0.92	0.84
	MD	Clean	71.77	62.70	326	51.27	78.39	0.348	0.157	0.93	0.83
		FN attack	70.87 (-0.90)	62.05 (-0.65)	343 (+5%)	50.32	78.27	0.357	0.169	0.00	0.00
		Daedalus	69.76 (-2.01)	60.77 (-1.93)	293 (-10%)	48.88	76.94	0.368	0.183	7.18	4.2
		Hijacking	68.66 (-3.11)	60.25 (-2.45)	399 (+22%)	48.69	76.28	0.372	0.189	0.93	1.11
		F&F	69.68 (-2.09)	60.87 (-1.83)	271 (-16%)	48.37	78.18	0.370	0.188	3.14	3.03
		Ours	57.75 (-14.02)	50.21 (-12.49)	937 (+187%)	45.80	56.47	0.387	0.189	0.92	0.92
MOT20	DS	Clean	85.8	69.40	1821	63.66	76.89	0.216	0.014	0.94	0.89
		FN attack	85.95 (+0.15)	69.28 (-0.12)	1576 (-13%)	63.46	76.85	0.216	0.014	0.00	0.00
		Daedalus	73.47 (-12.48)	62.73 (-6.67)	7778 (+327%)	62.25	64.52	0.227	0.023	1.48	1.28
		Hijacking	72.85 (-12.94)	61.22 (-8.18)	8061 (+442%)	61.97	61.60	0.228	0.022	0.89	0.84
		F&F	66.03 (-19.77)	58.29 (-11.11)	10990 (+503%)	61.05	56.93	0.235	0.040	3.19	3.02
		Ours	58.54 (-27.26)	51.11 (-18.29)	16669 (+815%)	58.34	46.00	0.264	0.054	0.92	0.88
	SS	Clean	86.63	69.97	943	64.05	77.52	0.215	0.014	0.94	0.89
		FN attack	86.55 (-0.08)	70.01 (+0.04)	978 (+3%)	64.26	77.38	0.212	0.015	0.00	0.00
		Daedalus	85.00 (-1.63)	69.02 (-0.94)	1850 (+96%)	63.91	75.63	0.214	0.018	1.48	1.16
		Hijacking	83.69 (-2.93)	68.34 (-1.62)	2658 (+181%)	63.86	74.23	0.215	0.018	0.89	0.85
		F&F	83.95 (-2.68)	68.38 (-1.59)	2411 (+155%)	63.66	74.48	0.216	0.021	3.19	2.87
		Ours	69.38 (-17.25)	58.90 (-11.07)	10014 (+961%)	62.95	55.97	0.223	0.024	0.92	0.88
	MD	Clean	88.93	73.017	828	66.46	81.45	0.177	0.019	0.94	0.89
		FN attack	88.00 (-0.93)	72.20 (-0.81)	1272 (+53%)	65.60	80.71	0.182	0.026	0.00	0.00
		Daedalus	85.06 (-3.87)	69.49 (-3.52)	2150 (+159%)	62.92	77.92	0.204	0.051	1.48	1.55
		Hijacking	85.30 (-3.62)	69.84 (-3.17)	2292 (+176%)	63.41	78.13	0.200	0.047	0.89	1.29
		F&F	84.40 (-4.53)	68.69 (-4.32)	1717 (+107%)	61.58	77.84	0.215	0.066	3.19	2.87
		Ours	66.10 (-22.83)	52.88 (-20.13)	16129 (+1847%)	55.34	51.74	0.267	0.127	0.92	0.88

Table 1: Experiment result of YOLOX and OSNet with maximum Mahalanobis distance threshold

creasing AssA, alongside a reduction in the detection metric DetA. As demonstrated by Tab. 1, unlike FN attack and Daedalus, BankTweak significantly affects performance by altering object IDs during attacks, with these changes persisting post-attack and resulting in a noticeable decline in IDF1 scores. Also, Hijacking and F&F attacks shift an object's position before the attack, creating a significant Mahalanobis distance from its original location after the attack, but increasing the Mahalanobis distance threshold (expanding the matching boundary) can neutralize the attack. BankTweak thrives when the matching boundary is expanded, ensuring the targeted object pair falls within this expanded boundary, enhancing performance.

Stealth. BankTweak induces ID switches by altering the feature banks of individual objects, effectively reducing accuracy without generating false alarms. Tab. 1 illustrates that BankTweak keeps the object count stable during an attack, as evidenced by the ρ^{Det} and ρ^{ID} metrics, closely aligning with the Clean scenario. For instance, detected object values on DS for Clean and BankTweak closely match (ρ^{Det} at 0.93 vs 0.92), similar to the tracked object values (ρ^{ID} remains at 0.82 for both). In contrast, Daedalus shows a marked effect on these metrics, with ρ^{Det} jumping from 0.93 to 7.18 and ρ^{ID} for tracked objects rising from 0.82 to 2.01, demonstrating a significant difference.

Applicability. In Tab. 1, against DS on MOT17, the Daedalus and F&F attacks reduce the IDF1 score by 5.3 and 7.75, respectively. In contrast, the reductions against SS are more modest, at 0.73 and 1.15, while for MD, the scores decrease by 2.01 and 2.09. These variations arise from the distinct matching strategies employed by each tracker. DS prioritizes matching newly tracked objects, which can result in previously tracked objects being incorrectly matched with false alarms during an attack. Conversely, SS and MD treat all tracked objects equally, reducing the likelihood of incorrect ID assignments. BankTweak, capable of attacking any tracker that employs a feature bank, operates effectively regardless of a tracker's specific matching procedures.

4.3 Ablation Study

Varying Mahalanobis Distance Threshold. In feature-based matching, trackers check if detected objects are within the tracked objects' *matching boundary*, set by a threshold λ^m. The Kalman filter models each tracked object's motion details (e.g., center, width, and height) using a chi-square distribution, represented by the probability density function $f(x)$. The matching boundary is where 95% of position values are concentrated around the distribution's mean within a λ^m distance. Figs. 4(a)–(c) show IDF1 variations as λ^m

Method	Clean			w/o Step 2			Ours		
Tracker	DS	SS	MD	DS	SS	MD	DS	SS	MD
IDF1	79.78	75.43	71.77	77.52	72.44	68.31	58.01	61.84	57.75

Table 2: Impact of Step 2 in BankTweak.

ϵ	4/255							16/255	
R^s	20	40	60	80	100	120	140	10	20
IDF-1	68.5	65.34	63.65	64.63	63.2	63.84	63.32	67.6	59.4

Table 3: Varying combinations of R^s and ϵ with $R^s = 10$.

increases (e.g., $\times 5$ indicates $\lambda^m \times 5$), noting that a higher λ^m expands the matching boundary, with "Max" occurring when $\int_0^{\lambda^m} f(x)dx = 1$. Clean and the FN attack experience minimal IDF1 changes with rising thresholds. Conversely, Daedalus and the F&F attack encounter an IDF1 increase and a notable decline in attack success as the threshold grows. BankTweak shows a high enhancement of the attack's effectiveness at higher thresholds, indicating its reliance on the matching boundary to facilitate ID switches.

Quantity of Allowable False Alarms. To evaluate attack effectiveness against stealth, we limited the number of false positives each attacker could generate. Figs. 4(d)–(f) plot IDF1 against the maximum number of objects for DS, SS, and MD, respectively, with D^{GT} representing the actual object count per attack frame and $2D^{GT}$, $3D^{GT}$, and $4D^{GT}$ denoting multiples of this number. The F&F attack typically generates four false positives per targeted object; instead of reducing these numbers, we constrained the targeted objects per frame. Clean and BankTweak do not generate additional objects, hence their IDF1 scores remain unaffected by the object limit. Daedalus and the F&F attack, however, show a notable IDF1 decrease with more false positives, although less significant than with BankTweak.

Impact of Step 2. As detailed in Fig. 3(c), Step 2 is crucial for BankTweak. Tab. 2 shows that ID switches occur less without Step 2, resulting in a smaller IDF1 reduction. For DS, omitting Step 2 results in a minor IDF1 drop (79.78 to 77.52), while including it reduces IDF1 by 21.77 points (79.78 to 58.01), underscoring Step 2's importance.

Number of Iterations. Tab. 3 shows the effect of varying BankTweak's R^s on IDF1. Increasing R^s leads to a greater reduction in IDF1, indicating that more iterations significantly enhance BankTweak's attack efficiency. Tab. 3 shows that increasing ϵ (in Eqs. (1) and (2)) dramatically reduces R^s needed to generate perturbations (which also reduces latency) while maintaining attack performance. For instance, with $\epsilon = 4/255$ (default), accuracy drops from 77.4 to 63.3. With $\epsilon = 16/255$, accuracy drops to 67.6 and 59.4 for $R^s = 10$ and $R^s = 20$, respectively.

5 Conclusion

This paper proposes BankTweak, a novel adversarial attack on multi-object trackers, targeting feature extractors during association to induce persistent ID switches. Our method remains robust under heightened Mahalanobis distance thresholds and does not depend on false alarms or motion prediction for effectiveness. We demonstrate the versatility of BankTweak across various combinations of detectors, feature extractors, and trackers. Comprehensive experiments on public datasets validate BankTweak's effectiveness and explore a diverse range of attack tactics.

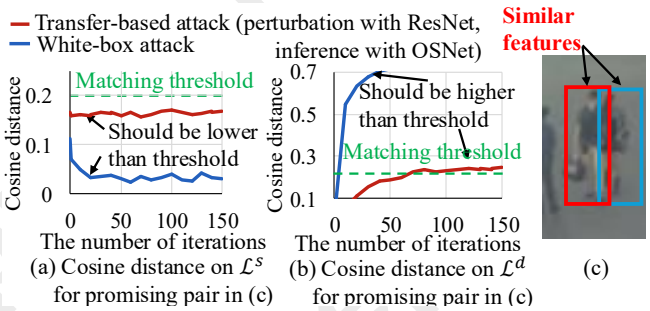


Figure 5: Experimental analysis of transfer-based attack for feature extractor

Limitation. While there has been significant research on black box attacks in classification problems [Mi *et al.*, 2023; Liang *et al.*, 2022], there has been little progress in MOT attacks due to its inherent challenge [Ding *et al.*, 2024]. Consequently, an MOT attack based on black-box methodologies has yet to be proposed. Although BankTweak is also a white-box attack, as illustrated in Fig. 5(c), a transfer-based black-box attack on BankTweak can be feasible when object pairs (A and B) for an ID switch have similar features. This is because generating similar features for A from B (or vice versa) is easier, and creating untargeted features (e.g., X and Y) is simpler than targeting generation. Figs. 5(a) and (b) show that, while less effective than a white-box attack, this approach meets the necessary thresholds for BankTweak in a transfer-based attack. Thus, a black-box attack will work if it finds object pairs with similar features to those of the deployed detector. Research on attacking MOT, targeting multiple objects simultaneously, is still in its early stages. Notable studies include the F&F attack and Hijacking, both white-box attacks, while BankTweak significantly addresses their limitations.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00520023, RS-2024-00438248, RS-2023-00250742). This research was supported by the MSIT(Ministry of Science and ICT), Korea under the ITRC(Information Technology Research Center) support program(IITP-2025-RS-2023-00259061) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

Contribution Statement

Woojin Shin and Donghwa Kang are co-first authors and contributed equally. Jinkyu Lee and Hyeongboo Baek are the corresponding authors.

References

[Chang *et al.*, 2018] Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2109–2118, 2018.

[Chen *et al.*, 2018] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *2018 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2018.

[Dai *et al.*, 2021] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a proposal classifier for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2443–2452, 2021.

[De Maesschalck *et al.*, 2000] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[Dendorfer *et al.*, 2020] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[Ding *et al.*, 2024] Xinlong Ding, Jiansheng Chen, Hongwei Yu, Yu Shang, Yining Qin, and Huimin Ma. Transferable adversarial attacks for object detection using object-aware significant feature distortion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[Du *et al.*, 2023] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023.

[Ge *et al.*, 2021] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Howard, 2017] Andrew G Howard. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[Jia *et al.*, 2020] Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, and Tao Wei. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *International Conference on Learning Representations (ICLR'20)*, 2020.

[Kalman, 1960] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[Kong *et al.*, 2020] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020.

[Kuhn, 1955] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[Liang *et al.*, 2022] Hongshuo Liang, Erlu He, Yangyang Zhao, Zhe Jia, and Hao Li. Adversarial attack and defense: A survey. *Electronics*, 11(8):1283, 2022.

[Lu *et al.*, 2017] Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv preprint arXiv:1712.02494*, 2017.

[Luiten *et al.*, 2021] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129:548–578, 2021.

[Luo *et al.*, 2021] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, 2021.

[Madry *et al.*, 2017] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[Meinhardt *et al.*, 2022] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8844–8854, 2022.

[Mi *et al.*, 2023] Jian-Xun Mi, Xu-Dong Wang, Li-Fang Zhou, and Kun Cheng. Adversarial examples based on object detection tasks: A survey. *Neurocomputing*, 519:114–126, 2023.

[Milan *et al.*, 2016] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[Ristani *et al.*, 2016] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pages 17–35. Springer, 2016.

[Sun *et al.*, 2020] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020.

[Wang *et al.*, 2021] Derui Wang, Chaoran Li, Sheng Wen, Qing-Long Han, Surya Nepal, Xiangyu Zhang, and Yang Xiang. Daedalus: Breaking nonmaximum suppression in object detection via adversarial examples. *IEEE Transactions on Cybernetics*, 52(8):7427–7440, 2021.

[Wojke *et al.*, 2017] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[Xie *et al.*, 2017] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017.

[Xu *et al.*, 2020] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, pages 665–681. Springer, 2020.

[Zhang *et al.*, 2021] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021.

[Zhang *et al.*, 2022] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 1–21. Springer, 2022.

[Zhou *et al.*, 2019] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3702–3712, 2019.

[Zhou *et al.*, 2020] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, pages 474–490. Springer, 2020.

[Zhou *et al.*, 2023] Tao Zhou, Qi Ye, Wenhan Luo, Kaihao Zhang, Zhiguo Shi, and Jiming Chen. F&f attack: Adversarial attack against multiple object trackers by inducing false negatives and false positives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4573–4583, 2023.

[Zhu *et al.*, 2021] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.