F&F Attack: Adversarial Attack against Multiple Object Trackers by Inducing False Negatives and False Positives

Tao Zhou[1] Qi Ye[1*] Wenhan Luo[2*] Kaihao Zhang[3] Zhiguo Shi[1] Jiming Chen[1]
[1]Zhejiang University [2]Sun Yat-sen University [3]Australian National University
{zhoutao2015, qi.ye, shizg, cjm}@zju.edu.cn, whluo.china@gmail.com, super.khzhang@gmail.com

Abstract

Multi-object tracking (MOT) aims to build moving trajectories for number-agnostic objects. Modern multi-object trackers commonly follow the tracking-by-detection strategy. Therefore, fooling detectors can be an effective solution but it usually requires attacks in multiple successive frames, resulting in low efficiency. Attacking association processes improves efficiency but may require model-specific design, leading to poor generalization. In this paper, we propose a novel False negative and False positive attack (F&F attack) mechanism: it perturbs the input image to erase original detections and to inject deceptive false alarms around original ones while integrating the association attack implicitly. The mechanism can produce effective identity switches against multi-object trackers by only fooling detectors in a few frames. To demonstrate the flexibility of the mechanism, we deploy it to three multi-object trackers (ByteTrack, SORT, and CenterTrack) which are enabled by two representative detectors (YOLOX and CenterNet). Comprehensive experiments on MOT17 and MOT20 datasets show that our method significantly outperforms existing attackers, revealing the vulnerability of the tracking-by-detection paradigm to detection attacks.

1. Introduction

As a common visual perception task, multi-object tracking (MOT) aims to build moving trajectories for number-agnostic objects. This requires the multi-object tracker to be capable of perceiving the birth, continuation, and termination of targets. To this end, most MOT methods [33, 22, 37, 34] follow the tracking-by-detection paradigm, working together with a detector [12, 38]. Given a new frame, the detector first finds all objects of interest. These detections are then associated with historical trajectories by various cues, like motion cues [3, 33, 37] and appearance cues [34, 29]. It has important applications in surveillance, autonomous driving, robotics [19], *etc*. Despite having been studied for decades and its importance, the robustness of MOT to attacks has just gained attention in recent years. In many other computer vision problems, numerous works have studied adversarial attacks against various visual perception tasks such as detection [28, 30], tracking [32, 14], semantic segmentation [30], etc since the vulnerability of deep learning models to adversarial examples is first investigated in [27].

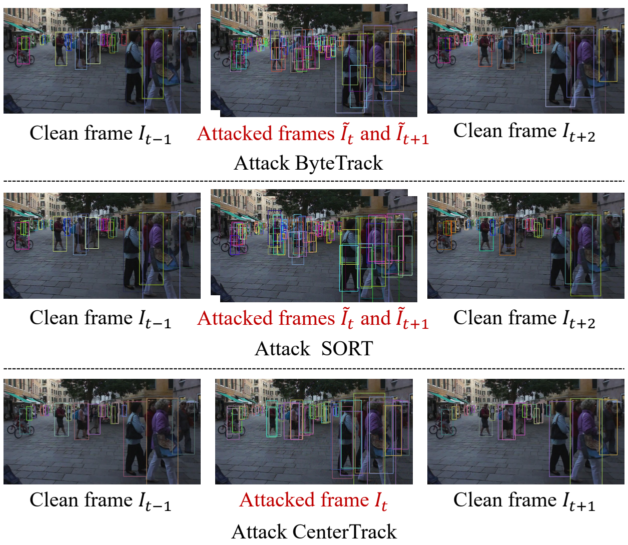


Figure 1: By erasing original detections and injecting deceptive false alarms around the original ones, our method misleads multi-object trackers to switch tracking identities of most targets after only attacking 1 or 2 frames. Bounding boxes with different colors represent different identities. Best viewed in color.

As detection is fundamental to tracking, attacking the detectors is a primary solution for the MOT attack. By utiliz-

[*] Corresponding author.

Qi Ye is with the College of Control Science and Engineering and the State Key Laboratory of Industrial Control Technology, Zhejiang University, and also with the Key Lab of CS&AUS of Zhejiang Province.

Project page: https://infzhou.github.io/FnFAttack/index.html ing the detection attacker, Daedalus [28], to produce dense false alarms, MOT is vulnerable to the attack in tracking targets of medium sizes (discussed in our experiments) and incurs a large number of identity switches. Also, the false negative attack, by making the object invisible to the model, has shown its effectiveness in single object tracking [32]. However, Daedalus shows poor effectiveness in attacking targets with large sizes, because the sizes of predicted boxes in Daedalus are extremely compressed to evade the non-maximum suppression (NMS) process. For false negative attack, successive attacks for a long period (e.g., 30 frames) are required to delete a trajectory as MOT usually adopts a "reserved period" [33, 34, 37] to avoid deleting a trajectory with occasional miss detections or short-term occlusions, which results in inefficient attacks.

In contrast, the Hijacking attacker [14], focuses on attacking the association process. It cheats the Kalman filter [15] inside the tracker by shifting the original detection box in a direction differing from the correct velocity, which could possibly trigger identity switches by attacking 1 frame. Despite the efficiency, it has three weaknesses. (1) The one-on-one mapping between shifted boxes and original boxes prefers independent perturbations for attacking each target, which does not hold when simultaneously attacking multiple targets in the scene. (2) It needs to repetitively forward the association component when solving the optimal shift (to check whether the shifted box is still correctly associated). (3) Poor performance in attacking multi-object trackers without Kalman filters.

To achieve both effective and efficient attacks, we propose the false positive and false negative attack (F&F attack) mechanism, which is a complementary integration of false alarm attack, false negative attack, and the idea of the association attack. The inspiration for our attack comes from the observation that in crowded scenes, severe occlusions between objects and frequent changes in visibility pose challenges in detection and association, leading to high probabilities of identity switches. Such challenging crowded scenes are simulated in our attack by erasing the original detection and injecting multiple deceptive false alarms around the original one. Specifically, three designs are adopted to increase the threat of false alarm attacks against multi-object trackers. (1) Instead of extremely compressing the size of each box to achieve a higher false alarm density, we trade off lower density for larger, more deceptive false alarms. (2) We erase the original detection to ensure one of the false alarms inherits the original identity, misleading association components to get incorrect estimations (e.g., velocity estimations). (3) We adopt a shifted and scaled design for false alarms to better evade NMS and further mislead association components. Note that, with the idea of association attack being implicitly integrated, our method attacks the multi-object tracker by fooling its detector component alone.

The F&F attack has the following advantages. (1) Effectiveness. The one-to-many design naturally benefits the simultaneous attack on multiple targets, as it better tolerates non-independent perturbations. (2) Simplicity. Our method efficiently fools multi-object trackers without accessing or forwarding association components. (3) Flexibility. The F&F attack is not specifically designed for attacking a certain multi-object tracker. Instead, attacks against trackers enabled by the same detectors share the same design. Besides, since detectors often share similar components, the F&F attack can be deployed on more detector families with minor modifications. This further broadens the scope of multi-object trackers at risk. As shown in Fig. 1, by perturbing a few frames, our method triggers high identity switching rates on several multi-object trackers, i.e., ByteTrack [33], SORT [3], and CenterTrack [37].

To summarize, our contributions are as follows:

- We propose a novel adversarial attack mechanism to efficiently cheat multi-object trackers by erasing the original detection, injecting deceptive false alarms, and integrating the association attack implicitly.
- We show the high flexibility of the mechanism by deploying it to different types of multi-object trackers.
- 24 experiments are constructed for four attackers attacking three modern trackers (CenterTrack, SORT, and ByteTrack) on two public datasets to study the different attacking behaviors and demonstrate the superiority of the proposed attack.

2. Related Work

2.1. Multi-Object Tracking

Given a video sequence, MOT builds moving trajectories for number-agnostic objects [20]. Most modern MOT methods [33, 22, 37, 34, 4, 8, 36] follow the tracking-by-detection paradigm. These methods can be roughly grouped into online ones [33, 37, 34, 29], where trajectories are extended at each time step, and offline ones [4, 8], which update trajectories after processing a batch of frames. In the tracking-by-detection paradigm, a detector [38, 24, 12] is first adopted to find objects of interest. Trackers then link these detections to historical trajectories by various cues. For example, the Kalman filter [15] is commonly used to estimate the motion cues. Zhou et al. [37] proposed to link targets by estimating their displacements across adjacent frames. FairMOT [34] and JDE [29] involved appearance embeddings to boost tracking performance. Recent transformer-based multi-object trackers [22, 26] used query embeddings to implicitly achieve detection and association, resulting in a new tracking paradigm.

Enabled by advanced detectors, modern multi-object trackers have achieved significant progress. However, the strong dependency on detectors may expose the vulnerability of MOT methods to detection attackers. In this paper, we reveal this risk by introducing a novel attack method, which achieves efficient attacks against multi-object trackers by solely fooling detectors.

2.2. Adversarial Attack

The vulnerability of deep learning models to adversarial examples was first investigated by Szegedy et al. [27]. After that, several methods, like FGSM[13] and PGD [21] were proposed to solve the perturbation efficiently. Recently, numerous works have studied adversarial attacks against various visual perception tasks such as detection [28, 30], tracking [32, 14], semantic segmentation [30], etc. Some studies [31, 6, 35, 10] further brought adversarial attacks to the physical world.

Close to MOT, several single object tracking (SOT) attackers [32, 7] were proposed based on different intuitions. Besides, there are some detection attackers aiming to trigger different misbehaviors, like missed detections [17], or false alarms [28], etc. However, these methods show limited effectiveness in attacking multi-object trackers due to the mission gap. Recently, Jia et al. [14] introduced an MOT attacker that focused on cheating the Kalman filter [15] inside the multi-object tracker. To maximize the effectiveness, [14] repetitively forwards the association component when solving the perturbation. Instead, we propose to attack multi-object trackers by fooling detectors alone, treating the association component as a black box.

3. False Negative and False Positive Attack

In this section, we first introduce the attack formulation in Sec. 3.1. Then, in Sec. 3.2, we explain the proposed attacking mechanism of misleading the trackers to switch tracking identities. Finally, in Sec. 3.3, we deploy the mechanism to attack different types of multi-object trackers.

3.1. Attack Formulation

Given a sequence of video frames $\mathbb{V} = \{I_1, \cdots, I_t, \cdots, I_N\}$, where $I_t \in \mathbb{R}^{W \times H \times 3}$, we add perturbations to a small subset of the frames, resulting in an attacked video $\tilde{\mathbb{V}} = \{I_1, \cdots, I_{t-1}, \tilde{I}_t, \cdots, \tilde{I}_{t+n-1}, I_{t+n}, \cdots, I_N\}$, where $I, \tilde{I}$ indicate original frames and attacked frames, respectively. The goal is to mislead multi-object trackers to switch tracking identities after the attack (i.e., since frame I_{t+n}).

A tracking-by-detection MOT system primarily consists of two parts, a detector module, and an association module. We deceive the MOT system by only attacking its detection module. To this end, we conduct targeted attacks on detectors with the aim of producing a targeted detection set that misleads the association and triggers identity switches to the maximum. In our experiments, we implement the detection attack under a *white-box* assumption but treat the association module as a *black box*.

Algorithm 1: F&F Attack

```
Data: Video subsequence $\mathbb{V} = [I_t, ..., I_{t+n-1}]$ to be
      attacked, object detector $D(\cdot)$.
Result: Attacked video subsequence
        $\tilde{\mathbb{V}} = [\tilde{I}_t, ..., \tilde{I}_{t+n-1}]$.
$\tilde{\mathbb{V}} = [\,]$
for $I$ in $\mathbb{V}$ do
    $\mathbb{D}^* \leftarrow D(I)$
    /* get targeted detections according to
       Sec. 3.3                                   */
    $\mathbb{D} \leftarrow$ get_targeted_detection($\mathbb{D}^*$)
    /* solve perturbations with Eq. 1-2           */
    $\tilde{I} \leftarrow$ solve_perturbation($I, \mathbb{D}, D(\cdot)$)
    $\tilde{\mathbb{V}}$.append($\tilde{I}$)
end
```

Formally, given a detector $D(\cdot|\theta)$ parameterized by θ, the original input image I, and the targeted detection set $\mathbb{D}$, the perturbations δ is optimized to minimize the targeted loss $\mathcal{L}_{\text{tgt}}$ ($\mathbb{D}$ and $\mathcal{L}_{\text{tgt}}$ are detailed in Sec. 3.3):

$$\delta = \underset{\delta, \|\delta\|_\infty \leq \epsilon}{\arg\min}\, \mathcal{L}_{\text{tgt}}(D(I + \delta|\theta), \mathbb{D}). \tag{1}$$

We adopt PGD [21] to iteratively solve the perturbations under ℓ_∞-norm constraint:

$$\delta^{r+1} = \text{clip}_{[-\epsilon,\epsilon] \cap [-I, 1-I]}(\delta^r + \alpha \text{sgn}(\nabla_\delta \mathcal{L}_{\text{tgt}}(D(I+\delta|\theta), \mathbb{D}))), \tag{2}$$

where ϵ limits the maximum perturbation for each pixel, α controls the step length of each iteration, ∇ indicates the gradient operation, and $\text{sgn}(\cdot)$ extracts the sign of gradients. Perturbations are clipped to meet the ℓ_∞-norm constraint and to ensure the perturbed input is within [0, 1]. We use zero initialization for δ^0, and obtain the final perturbation δ^R after R iterations.

Alg. 1 shows the pipeline of our method. To conduct attacks on frame I_t, we first get the original detection set $\mathbb{D}^*$ by forwarding the detector $D(\cdot)$ with the clean image I_t. Then we get targeted detections by erasing original detections and injecting deceptive false alarms according to the design specified in Sec. 3.3. Finally, the perturbation δ is solved by PGD [21] aiming at minimizing Eq. 1. As we do not access or forward association components when solving perturbations, our attack pipeline is concise.

3.2. F&F Attack Mechanism

The main idea of the mechanism is two-fold. (1) F&F injects false alarms into the attacked frame $\tilde{I}_t$, letting them

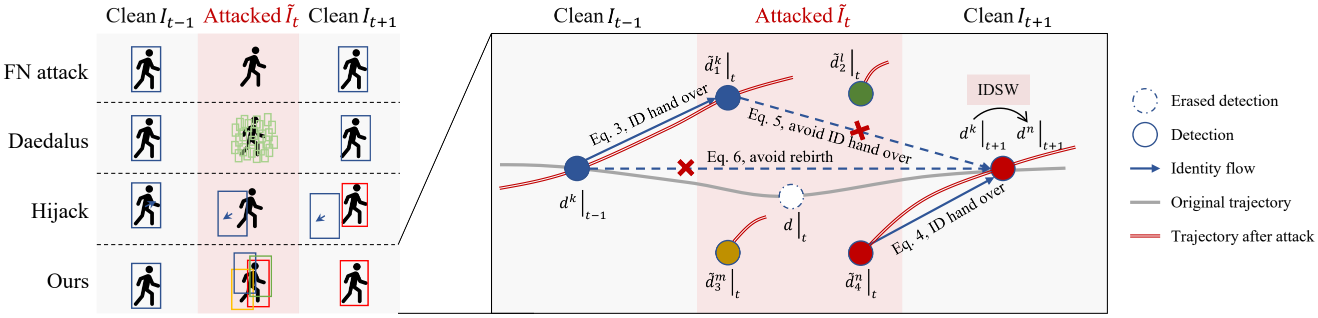


Figure 2: Diagrams of existing attackers and an example of our method. In the example, circles filled with different colors identify detections with different tracking identities. To attack the target $d^k|_{t-1}$ with tracking identity k, we erase its original detection $d|_t$ in the attacked frame $\tilde{I}_t$ and inject 4 deceptive false alarms $\{\tilde{d}_1|_t, \tilde{d}_2|_t, \tilde{d}_3|_t, \tilde{d}_4|_t\}$ around $d|_t$. With such perturbation at time step t, the tracker may link one of the false alarms to the existing trajectory with identity k (resulting in $\tilde{d}_1^k|_t$) and spawn 3 new trajectories for the remaining false alarms with new identities, l, m, n, respectively. Then, at time step $t+1$, the 4 trajectories with identities k, l, m, n compete for the detection $d|_{t+1}$ in the unattacked frame I_{t+1}. An identity switch (IDSW) occurs if one of the newly spawned trajectories (i.e., with the identity of n in the figure) wins the competition.

compete for the correct tracking ID and prevent the ID from being correctly propagated from I_{t-1} to I_{t+1}. (2) F&F erases the correct detections in the attacked frame $\tilde{I}_t$, ensuring that the ID in frame I_{t-1} is inherited by one of the false alarms.

To elaborate on the mechanism, we construct an example in Fig. 2. For simplicity, we analyze the case of tracking a single target and we neglect the probationary period. We denote each original detection at time step t as $d|_t$. It is further denoted by $d^k|_t$ if it inherits the tracking identity k after association. Similarly, false alarms are denoted by $\tilde{d}_i|_t$ where i indicates the detection index. Assuming that trackers conduct associations in a greedy manner, then false alarms with indexes of $a = \arg\max_i(sim(d^k|_{t-1}, \tilde{d}_i|_t))$ and $b = \arg\max_i(sim(d^k|_{t+1}, \tilde{d}_i|_t))$ transfer the identity k in time step $t-1 \to t$ and $t \to t+1$, respectively, where $sim(\cdot,\cdot)$ indicates the similarity measurement (e.g., intersection over union (IoU)) used in association. In Fig. 2, we have $a = 1$ and $b = 4$. One identity switch is triggered if Eq. 3 to Eq. 6 are satisfied:

$$sim(d^k|_{t-1}, \tilde{d}_a|_t) > \tau, \tag{3}$$

$$sim(d|_{t+1}, \tilde{d}_b|_t) > \tau, \tag{4}$$

$$a \neq b, \tag{5}$$

$$sim(d|_{t+1}, \tilde{d}_b|_t) > sim(d|_{t+1}, d^k|_{t-1}), \tag{6}$$

where τ indicates the prior similarity threshold inside trackers above which the association is accepted. Eq. 3 to Eq. 5 ensure that separate false alarms (with indexes of a and b) transfer the identity in time step $t-1 \to t$ and $t \to t+1$, respectively. Eq. 6 avoids the rebirth of the original trajectory.

Notice that the attack mechanism does not explicitly access the similarity measurement inside the tracker (i.e., the function $sum(\cdot,\cdot)$). Instead, it maximizes the probability of Eq. 3 to Eq. 6 being met through the design of targeted detections $\mathbb{D}$ (detailed in Sec. 3.3).

3.3. Targeted Attack Design

3.3.1 Target-Size-Aware Shift Strategy

Two conflicts pose challenges in the design of the F&F attack. Firstly, association modules and detector modules have conflicted preferences for box overlap. Association modules favor greater overlaps (higher similarity), but heavily overlapped boxes are eliminated by NMS. Secondly, the F&F attack requires dense isolated responses on the confidence prediction map (in order to erase the original detection and inject false alarms near the original ones), which conflict with the spatial smoothness of network predictions.

To address these challenges, we propose a target-size-aware shift strategy. In specific, we set targeted detections $\mathbb{D}$ by replacing each original detection with γ (e.g., $\gamma = 4$) shifted false alarms so that (1) leaves space for smooth variations in the confidence prediction map and (2) makes the overlaps between false alarms and original detections deceptive (i.e, neither be dropped by NMS nor be rejected in association). Each false alarm is shifted by $(\kappa w, \kappa h)$ away from the original one in γ different directions, where κ controls the overlaps between shifted false alarms, w and h are the width and height of original detection, respectively. Another benefit of this design is its ability to perturb the state estimation (e.g., velocity estimation) of the association module.

Additionally, in order to further reduce the overlap between false alarms and to further mislead association modules, the height and width of false alarms are respectively scaled to sh and sw, where the hyperparameter $s \in [0, 1]$. By involving translation and scaling, F&F integrates the idea of association attack. We discuss the settings of (κ, s) in Sec. 4.3.

3.3.2 Perturbation Solving

Targeted Loss for Attacking ByteTrack. The targeted loss is designed as

$$\mathcal{L}_{\text{tgt}}^{\text{YOLOX}} = \mathcal{L}_{\text{obj}} + \lambda \mathcal{L}_{\text{L1}}, \tag{7}$$

where $\mathcal{L}_{\text{obj}}$ and $\mathcal{L}_{\text{L1}}$ are inherited from the training loss of YOLOX [12], supervising the classification task and the regression task, respectively. We use $\lambda = 1$.

Before calculating the loss, a detector typically needs a strategy to assign anchors to ground truth. Originally, YOLOX adopts a dynamic k assignment policy [11] during training, based on the intuition that the number of positive anchors should differ across targets with different sizes and occlusion states. However, using this biased assignment mechanism when minimizing the targeted attack loss $\mathcal{L}_{\text{tgt}}$ leads to imbalanced attack performance across targets and limits the number of positive anchors assigned to each target. We address these two limitations by fixing the k to a reasonably large value (e.g., $k = 16$).

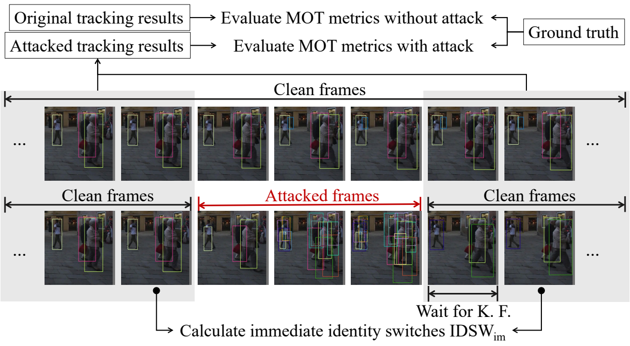
As SORT [3] and ByteTrack are enabled by the same detector in our experiments, the F&F attack on SORT thus shares the same design of attacking ByteTrack.

Targeted Loss for Attacking CenterTrack. To showcase the flexibility of the F&F attack, we deploy it to attack CenterTack [37], which is enabled by another representative detector (CenterNet [38]). Given the targeted detection set $\mathbb{D}$, we render a center heatmap following [38]. Similar to attacking ByteTrack, the targeted loss for attacking CenterTrack is designed as

$$\mathcal{L}_{\text{tgt}}^{\text{CenterTrack}} = \mathcal{L}_{\text{p}} + \lambda_{\text{s}} \mathcal{L}_{\text{s}}, \tag{8}$$

where $\mathcal{L}_{\text{p}}$ supervises the heatmap prediction and $\mathcal{L}_{\text{s}}$ supervises the box regression. Both of them are inherited from the training loss of CenterTrack [37]. We set $\lambda_{\text{s}} = 0.1$.

Considering that detectors often share similar components (e.g., a classification branch and a localization branch), the F&F attack can be easily deployed to attack more detector families with minor modifications, thereby expanding the scope of threatened MOT systems.



Original tracking results
Evaluate MOT metrics without attack
Ground truth
Attacked tracking results
Evaluate MOT metrics with attack
Clean frames
...
Clean frames
Attacked frames
Clean frames
Wait for K. F.
Calculate immediate identity switches IDSW$_{\text{im}}$

Figure 3: The attack performance evaluation contains two parts. (1) We evaluate the MOT metrics of newly assembled sequences containing unattacked frames from each result (in gray). (2) Attack success rate is defined as the ratio of immediate identity switches IDSW$_{\text{im}}$ right after the attack.

4. Attack Evaluation

4.1. Experiment Methodology

Evaluation Metrics. As shown in Fig. 3, our evaluation contains two parts. First, given the track results $\mathbb{L}$ and $\tilde{\mathbb{L}}$ of the clean sequence and the attacked sequence, respectively, we assemble new sequences by extracting the parts of unattacked frames from each result to evaluate the MOT metrics, including CLEAR [2], IDF1 [25], and HOTA [18]. Note that, we exclude the attacked frames in the evaluation to avoid distorted or meaningless association metrics. Second, in order to provide a straight view of attack performance, we calculate the ratio of immediate identity switches IDSW$_{\text{im}}$ right after the attack. Considering that the Kalman filter may require several time steps to catch up with rapidly changing observations (detections), we leave one additional clean frame waiting for the Kalman filter before counting the IDSW$_{\text{im}}$ (otherwise, some targets may be regarded as missed detections instead of experiencing identity switching).

Datasets. We conduct experiments on two widely used pedestrian tracking datasets, MOT17 [23] and MOT20 [9]. MOT17 is characterized by various viewpoints and different target sizes. MOT20 is characterized by high density and heavy occlusion. Since we need ground truth to evaluate the MOT metrics with and without attacks, all experiments are conducted on the training splits of two datasets. Following common practices [33, 34, 37], we split each training sequence into two halves, using the first half for training models and the rest for evaluating attacks. To enrich the sequence, we split each evaluation sequence into segments every 30 frames, resulting in 83 segments on MOT17 and 148 segments on MOT20. For each segment, we only attack once, starting from the 5th frame (instead of starting from the beginning frame) for practice considerations.

Table 1: Detailed experimental settings.

Tracker	ϵ	α	#iter	P	τ_{NMS}	τ_{track}	#Fm.
ByteTrack	4/255	1/255	150	1	0.7	0.1	3
SORT	4/255	1/255	150	1	0.7	0.3	3
CenterTrack	8/255	1/255	60	0	-	-	1

Implementation Details. We evaluate our attack against three multi-object trackers, CenterTrack [37], ByteTrack [33], and SORT [3]. Detailed experimental settings are provided in Table 1, where #iter is the number of iterations, P is the probation period, τ_{NMS} is the NMS threshold, τ_{track} is the IoU threshold below which an association is rejected, and #Fm. is the number of attacked frames. For CenterTrack and ByteTrack, we adopt their official implementations and follow the same tracking settings as the authors. For SORT, we adopt the implementation from ByteTrack, which is enabled by the YOLOX detector. We set greater ϵ when attacking CenterTrack because it stacks two frames as input but we only add perturbations to a single frame. Besides, we find trackers using standard NMS operations slightly benefit from more iterations. We set #iter=150, γ=4, κ=0.2, and s=0.8 by default when attacking ByteTrack and SORT. We provide discussions of better effectiveness and fewer iterations in the ablation study. More implementation details can be found in the supplement.

4.2. Compare to Existing Attackers

In Table 2, we compare our method with three baseline attackers: (1) False Negative attacker [32, 17], which is usually used in attacking detectors and single-object trackers, aiming to make the target invisible to the model; (2) Daedalus [28], a detection attacker, which induces dense false alarms by raising the confidence predictions along with compressing the predicted sizes of boxes; (3) Hijacking [14], an MOT attacker, which is designed to cheat the Kalman filter inside the tracker. Due to the space limit, we do not list hybrid metrics (i.e., HOTA and MOTA) considering that they can be calculated by $\text{HOTA} = \sqrt{\text{DetA} \cdot \text{AssA}}$ and $\text{MOTA} = 1 - \text{FN} - \text{FP} - \text{IDSW}$.

According to Table 2, we have the following observations and analysis.

Effectiveness of Our Attack Mechanism. Shown by the higher attack success rate IDSW_{im} and the greater decline in association metrics including AssA, IDF1, and IDSW, our method significantly and consistently outperforms baseline attackers. It might be noticed that the decline in detection metrics (e.g., DetA, FN, and FP) is less remarkable. This is due to the exclusion of attacked frames during the evaluation, as depicted in Fig. 3.

Superiority to False Negative Attack and Daedalus. The false negative attack (denoted by "FN Attack") shows poor effectiveness because multi-object trackers are designed to resist occasional miss detections. The Daedalus (false alarm attack) shows effectiveness in attacking ByteTrack on the MOT20 dataset, where the majority of the targets are of medium size. However, its effectiveness remarkably decreases when applied to the MOT17 dataset, which contains a variety of object sizes. This validates the sensitivity of Daedalus to target sizes. In contrast, the consistent and better performance of our method suggests that the weakness of Daedalus is overcome in our design. Another observation is that SORT shows robustness against Daedalus, mainly because it adopts a higher association threshold ($\tau_{\text{track}} = 0.3$) by default. This higher threshold rejects the association between trajectories and small-sized false alarms generated by Daedalus.

Superiority to Hijacking. The Hijacking attack performs worse on the MOT20 dataset compared to that on MOT17, while our method exhibits an opposite trend in performance. Considering the higher target density in the MOT20 dataset, this highlights the advantage of our one-to-many design. We provide further visualization analysis in the supplement. As the Hijacking attack is specifically designed to attack Kalman filters, our method significantly outperforms Hijacking when the Kalman filter is absent (in CenterTrack), validating the high flexibility of our method.

Impacts of Different NMS Mechanisms. CenterTrack adopts a 3×3 max pooling operation on the confidence heatmap as an alternative to classic NMS operation, leading to the defacto detection set $\hat{\mathbb{D}}$ (obtained by forwarding the model with the perturbated image) being controllable. With $\gamma = 4$, we inject 4 false alarms for each original detection in CenterTrack, resulting in an expected value of 75% for IDSW_{im}, which is close to the experimental results in Table 2. In contrast, when attacking YOLOX-enabled trackers, the defacto detections $\hat{\mathbb{D}}$ are observed to differ from $\mathbb{D}$ due to the lower controllability of the classic NMS operation. Benefitting from our modifications in the label assignment strategy, this low controllability yields a higher attack performance than expected (i.e., $\text{IDSW}_{\text{im}} > 75\%$ when attacking ByteTrack and SORT).

4.3. Ablation Study

As shown in Table 3, we ablate our design on attacking ByteTrack as an example. Two untargeted attackers are adopted as baselines of our targeted detection design: the FN attacker, and the FP attacker, where miss detections or false alarms are blindly induced. The poor performance of untargeted attackers validates the effectiveness of our targeted design. Besides, the attack performance decreases without fixing the k in label assignment (denoted by "w/o fixed k" in Table 3) due to the reduction in the numbers of deceptive false alarms and the unbalanced attack performance between objects with different sizes. Removing the regression loss $\mathcal{L}_{\text{L1}}$ (denoted by "w/o $\mathcal{L}_{\text{L1}}$" in Table 3) also harms the attack effectiveness on ByteTrack. This is mainly

Table 2: Attack performance comparison.

Dataset	Tracker	Attacker	#Fm.	IDSW$_{\text{im}}$↑	DetA↓	AssA↓	IDF1↓	FN(%)↑	FP(%)↑	IDSW(%)↑	IDs↑
MOT17	CenterTrack	Clean	-	-	56.61	82.61	80.11	29.07	2.76	0.23	1615
		FN Attack	1	1.05%	56.43	82.34 (-0.27)	79.78 (-0.33)	29.66	2.48	0.25	1614
		Daedalus	1	6.27%	56.50	80.80 (-1.81)	78.96 (-1.15)	28.90	3.34	0.43	1809
		Hijacking	1	25.12%	56.42	74.68 (-7.93)	75.82 (-4.29)	29.45	2.70	0.81	1712
		Ours	1	**74.38%**	56.23	57.48 (-**25.13**)	64.93 (-**15.18**)	28.95	3.40	2.89	2704
	ByteTrack	Clean	-	-	66.67	85.50	87.58	17.92	3.88	0.18	1739
		FN Attack	3	3.45%	66.34	84.57 (-0.93)	86.78 (-0.80)	18.26	3.99	0.36	1755
		Daedalus	3	51.21%	61.90	69.28 (-16.22)	77.07 (-10.51)	18.39	6.03	2.57	2768
		Hijacking	3	68.17%	65.03	66.34 (-19.16)	77.28 (-10.30)	19.02	3.94	2.14	2218
		Ours	3	**85.00%**	63.83	60.63 (-**24.87**)	73.76 (-**13.82**)	17.39	5.05	3.13	3105
	SORT	Clean	-	-	66.72	84.15	86.44	16.15	6.21	0.84	2242
		FN Attack	3	4.02%	66.58	83.50 (-0.65)	85.89 (-0.55)	16.39	6.21	0.98	2261
		Daedalus	3	8.48%	66.55	82.03 (-2.12)	84.53 (-1.91)	16.05	6.58	1.62	2725
		Hijacking	3	68.03%	65.91	66.79 (-17.36)	76.04 (-10.40)	16.92	6.17	2.98	3077
		Ours	3	**78.29%**	65.67	63.67 (-**20.48**)	73.89 (-**12.55**)	16.24	6.58	3.81	3686
MOT20	CenterTrack	Clean	-	-	62.56	82.29	86.46	20.57	2.91	0.15	19268
		FN Attack	1	0.62%	61.82	81.54 (-0.75)	85.60 (-0.86)	22.04	2.44	0.17	19189
		Daedalus	1	18.36%	61.68	75.40 (-6.89)	81.74 (-4.72)	20.94	3.73	0.86	22841
		Hijacking	1	37.09%	61.90	68.77 (-13.52)	78.78 (-7.68)	20.66	3.83	1.20	21733
		Ours	1	**75.09%**	60.18	52.66 (-**29.63**)	65.46 (-**21.00**)	18.60	8.26	4.44	41685
	ByteTrack	Clean	-	-	71.64	85.42	92.77	10.67	2.32	0.11	20106
		FN Attack	3	0.35%	71.48	85.35 (-0.07)	92.63 (-0.14)	11.00	2.19	0.11	20074
		Daedalus	3	80.96%	67.75	62.74 (-22.68)	78.67 (-14.10)	11.25	3.53	2.86	35684
		Hijacking	3	57.97%	69.98	66.87 (-18.55)	82.89 (-9.88)	11.63	2.62	2.02	22975
		Ours	3	**88.56%**	69.54	61.00 (-**24.42**)	78.25 (-**14.52**)	10.14	3.26	3.09	37256
	SORT	Clean	-	-	72.51	85.44	93.14	9.58	2.88	0.21	22022
		FN Attack	3	0.78%	72.50	85.39 (-0.05)	93.10 (-0.04)	9.62	2.86	0.21	22010
		Daedalus	3	6.32%	72.34	84.10 (-1.34)	92.10 (-1.04)	9.59	3.06	0.44	23883
		Hijacking	3	58.92%	71.71	68.87 (-16.57)	83.27 (-9.87)	10.19	3.07	2.23	28950
		Ours	3	**87.59%**	71.09	61.49 (-**23.95**)	77.76 (-**15.38**)	9.58	3.14	3.47	40376

Table 3: Ablation study on the MOT17 dataset.

Tracker	Ablations	IDSW(im.)↑	ΔAssA↓	ΔIDF1↓
ByteTrack	FN attack	3.45%	-0.93	-0.80
	FP attack	50.17%	-17.74	-11.57
	w/o fixed k	71.16%	-19.84	-11.04
	w/o $\mathcal{L}_{\text{L1}}$	64.05%	-18.02	-10.5
	Full method	**85.00%**	**-24.87**	**-13.82**

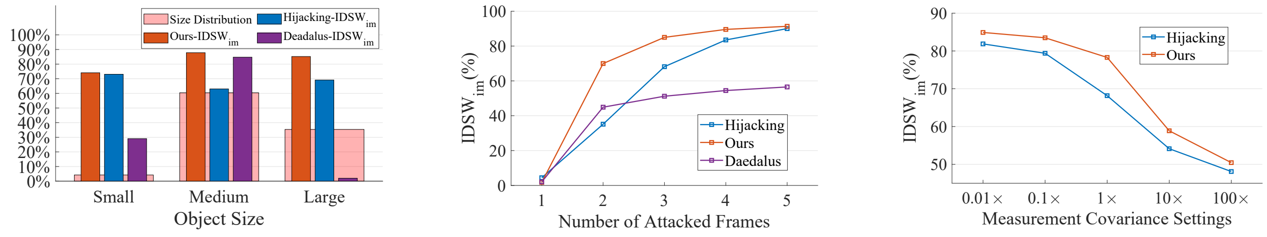
because ByteTrack performs association based on box representation. The regression loss $\mathcal{L}_{\text{L1}}$ ensures that deceptive false alarms are injected at expected locations with expected sizes.

To further highlight the advancement of our method, we analyze the attack performance under different experimental settings. If not specified, experiments are conducted on attacking ByteTrack.

Object sizes. We investigate the attack performance against different object sizes in Fig. 4a, where we follow the definition of object sizes from the COCO dataset [16] and we use red bars to indicate the object size distribution in the validation split of MOT17 dataset. It can be observed that Daedalus has poor effectiveness on small-sized objects and extremely poor effectiveness on large-sized objects, while the Hijacking attack and our method are not sensitive to object size.

Number of attacked frames. A probationary period is adopted by some trackers [33, 34, 3] when spawning a new trajectory in order to prevent tracking of occasional false positives. Due to the probationary period within ByteTrack and SORT, we set the default number of attacked frames to 3 when attacking these two trackers. As an ablation, we evaluate the attack performance by attacking {1, 2, 3, 4, 5} frames in Fig. 4b. Our method shows a significant increase in performance when the attacked frame number (i.e., 2) exceeds the probationary period (i.e., 1) and shows constantly better performance compared to baseline attackers. After each attack, the Hijacking attacker checks whether each target has been successfully attacked and only attacks the unattacked targets in subsequent frames, leading to a decreasing attack difficulty. However, our method does not perform this check because we promise not to access the association module.

Covariance of measurement noise. The prior measurement covariance settings in the Kalman filter affect the effectiveness of the attack. A lower measurement covariance results in the filter placing more trust in the measurements (i.e., detections), while a higher one makes the filter more confident in the predictions. We conduct



(a)

(b)

(c)

Figure 4: Attack success rate with regard to (a) object sizes (b) the number of attacked frames (c) measurement noise covariance settings of the Kalman Filter.

Table 4: Attack effectiveness with different shift (κ) and scale (s) settings on the MOT17 dataset.

κ	s	IoU$_1$	IoU$_2$	IDSW$_{\mathrm{im}}$(ByteTrack)	IDSW$_{\mathrm{im}}$(SORT)
0.1	0.4	0.16	0.33	87.73%	24.18%
0.1	0.6	0.36	0.50	81.73%	80.82%
0.1	0.8	0.64	0.60	73.58%	72.42%
0.2	0.4	0.16	0	94.43%	18.91%
0.2	0.6	0.36	0.20	93.82%	**81.04%**
0.2	0.8	0.43	0.33	85.00%	78.29%
0.3	0.4	0.16	0	**95.02%**	11.90%
0.3	0.6	0.23	0	92.62%	48.77%
0.3	0.8	0.28	0.14	83.50%	71.13%

experiments on SORT where a classic Kalman filter is adopted, supporting the independent adjustment of observation covariance. The original measurement covariance is scaled by $\{0.01\times, 0.1\times, 1\times, 10\times, 100\times\}$ times in experiments. As shown in Fig. 4c, our method consistently outperforms the Hijacking attack under a wide range of measurement covariance settings.

Better Effectiveness. For the sake of generalization, we use $\kappa = 0.2$ and $s = 0.8$ for both attacking ByteTrack and attacking SORT. However, setting specific values for attacking different trackers can lead to better effectiveness. Table 4 details the attack performances under different (κ, s) combinations. In addition, we use IoU$_1$ to denote the IoU between the original detection and one of its shifted false alarms, and use IoU$_2$ to denote the maximum IoU between the shifted false alarms that belong to the same original detection. According to Table 4, we have two observations. (1) Setting κ and s to make IoU$_1$ slightly higher than the association threshold τ_{track} leads to the best attack performance. (2) By involving translation and scaling in our design, the false alarm boxes effectively evade the NMS (i.e., IoU$_2 < \tau_{\text{NMS}}$).

Fewer Iterations. We report the attack performance using fewer iterations in Table 5, where we set κ=0.3 and s=0.4 for F&F. F&F achieves an attack success rate of 69.5% within 10 iterations. To further reduce the time cost of each iteration, an asynchronous attack (attacking a small subset of targets within the image at a time) could be implemented on the engineering side.

Table 5: Experiments of attacking ByteTrack on MOT17 dataset with fewer iterations.

Method	Attack Success Rate IDSW$_{\mathrm{im}}$(%) ↑				
	#iter=2	#iter=4	#iter=6	#iter=8	#iter=10
Daedalus	1.4	10.6	20.6	27.9	36.0
Hijacking	**7.0**	15.9	22.7	29.5	36.1
Ours	5.5	**26.5**	**47.5**	**62.0**	**69.5**

Table 6: Effectiveness under common defense algorithms.

	No Defense	CJ	GN	SS	AT
IDSW$_{\mathrm{im}}$(%) ↑	91.4	90.8	86.9	75.8 (+EoT)	82.0 (ℓ_∞, #iter⇑)

4.4. Discussion

Limitation. Though our method achieves advanced effectiveness and efficiency in attacking motion-based multi-object trackers (e.g., ByteTrack, SORT, CenterTrack, etc.), the effectiveness may degrade when attacking some re-identification-based trackers due to the natural limitation of fooling detectors alone. The major challenge lies in the smooth updating of appearance embeddings inside the tracker, which leads to the violation of Eq. 5 and Eq. 6. We elaborate on this limitation in the supplement by deploying the F&F attack to attack FairMOT [34], where we also provide suggestions for enhancing the F&F attack. Another limitation of F&F is that the current version remains in the digital domain. We leave the implementation in the physical world as future work.

Defense. Based on the limitation mentioned above, strengthening the smoothness constraint in state estimation (e.g., velocity, appearance) can reduce the effectiveness of F&F, but it may also impact clean tracking performance. Checking abrupt changes in the number of detections could be another feasible way. Besides, Table 6 summarizes the effectiveness of F&F under some common defense algorithms, where we deploy the F&F to attack ByteTrack on the MOT17 dataset and set κ=0.3, s=0.4, #iter=30, and ℓ_∞=4/255 by default. F&F is found to be robust to color jitter **CJ** (where the jitter for each color channel is inde-

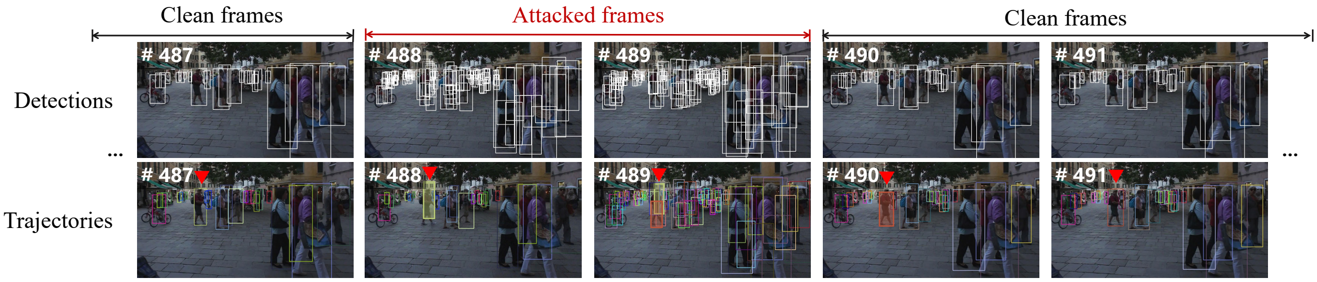


Figure 5: Qualitative results of deploying F&F to attack ByteTrack. We list the detection results in the first line and the association results in the second line. Tracking identities are coded by color. The target highlighted by red triangles validates our hypothesis presented in Fig. 2. For more results and analysis please refer to the supplement.

pendently sampled within ±50/255) and minor Gaussian noise **GN** (σ=2/255). If facilitated by the commonly used EoT [1] technique, F&F becomes robust to local spatial smoothing **SS** (e.g., 3x3 average smoothing). Finally, F&F keeps effective against adversarially trained **AT** [5] models if larger bounds and greater iteration numbers are allowed (e.g., ℓ_∞=64/255, #iter=80).

5. Conclusion

In this paper, we propose a novel attack mechanism, F&F attack, which attacks multi-object trackers by solely conducting detection attacks while integrating the association attack implicitly. The challenging crowded scenes are simulated by erasing original detections along with injecting deceptive false alarms, finally misleading trackers to switch tracking identities. The flexibility of the proposed mechanism is demonstrated by deploying it to attack three multi-object trackers, ByteTrack, SORT, and CenterTrack, which are enabled by detectors using different NMS mechanisms. The advanced performance of our method is witnessed in comprehensive experiments on MOT17 and MOT20 datasets. We hope that the vulnerability of MOT methods to detection attacks revealed in this paper can inspire the MOT community.

Acknowledgements

This work was supported in part by NSFC under Grants (62088101, 62103372, 62233013), the Key Research and Development Program of Zhejiang Province (2021C03037), U21A20456, Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515012839), and Shenzhen Science and Technology Program (No. JSGG20220831093004008).

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018. 9

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 5

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016. 1, 2, 5, 6, 7

[4] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6247–6257, 2020. 2

[5] Pin-Chun Chen, Bo-Han Kung, and Jun-Cheng Chen. Class-aware robust adversarial training for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10420–10429, 2021. 9

[6] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 52–68. Springer, 2019. 3

[7] Xuesong Chen, Xiyu Yan, Feng Zheng, Yong Jiang, Shu-Tao Xia, Yong Zhao, and Rongrong Ji. One-shot adversarial attacks on visual tracking with dual attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10176–10185, 2020. 3

[8] 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. 2

[9] 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. 5

[10] Ranjie Duan, Xiaofeng Mao, A Kai Qin, Yuefeng Chen, Shaokai Ye, Yuan He, and Yun Yang. Adversarial laser

beam: Effective physical-world attack to dnns in a blink. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16062–16071, 2021. 3

[11] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 303–312, 2021. 5

[12] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 1, 2, 5

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 3

[14] Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, and Tao Wei Wei. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *International Conference on Learning Representations (ICLR'20)*, 2020. 1, 2, 3, 6

[15] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 2, 3

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 7

[17] Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv preprint arXiv:1712.02494*, 2017. 3, 6

[18] 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. 5

[19] Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1317–1332, 2019. 1

[20] 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. 2

[21] 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. 3

[22] 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. 1, 2

[23] 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. 5

[24] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[25] 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. 5

[26] 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. 2

[27] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 3

[28] 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. 1, 2, 3, 6

[29] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 107–122. Springer, 2020. 1, 2

[30] 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. 1, 3

[31] 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 16*, pages 665–681. Springer, 2020. 3

[32] Bin Yan, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 990–999, 2020. 1, 2, 3, 6

[33] 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. 1, 2, 5, 6, 7

[34] 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. 1, 2, 5, 7, 8

[35] Yiqi Zhong, Xianming Liu, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15345–15354, 2022. 3

[36] Tao Zhou, Wenhan Luo, Zhiguo Shi, Jiming Chen, and Qi Ye. Apptracker: Improving tracking multiple objects in low-

frame-rate videos. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6664–6674, 2022. 2

[37] 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. 1, 2, 5, 6

[38] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 2, 5