Supplementary Materials for FnF 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

This document contains additional material for the main submission. Sec. A provides further implementation details for our method. Sec. B details the weakness of the one-to-one design [3] by a visualization example. Sec. C elaborates on the limitation of the proposed F&F attacker by deploying it to attack FairMOT [7], where we also provide suggestions for enhancing the F&F attacker. Sec. D contains further qualitative analysis of our method.

A. Further Implementation Details

We deploy the proposed F&F attacker on four multi-object trackers in our experiments, including ByteTrack [6], SORT [1], CenterTrack [8], and FairMOT [7]. We adopt the official implementations and tracking configurations for ByteTrack[1], CenterTrack[2], and FairMOT[3]. As for SORT, we use the implementation from ByteTrack, which is enabled by the YOLOX [2] detector. Detailed tracking configurations are summarized in Table A1, where τ_{NMS} is the NMS threshold, τ_{track} is the IoU threshold below which an association is rejected, P is the probation period. CenterNet-enabled trackers, like CenterTrack and FairMOT, use the max pooling operation as an alternative to classic NMS operations. Besides, CenterTrack and FairMOT do not use explicit IoU thresholds to gate associations.

Table A1: Detailed tracking configurations.

Tracker	Detector	NMS	τ_{NMS}	τ_{track}	P
ByteTrack	YOLOX	classic NMS	0.7	0.1	1
SORT	YOLOX	classic NMS	0.7	0.3	1
CenterTrack	CenterNet	max pooling	-	-	0
FairMOT	CenterNet	max pooling	-	-	1

A.1. Detailed Design for Attacking CenterNet-Enabled Trackers

Compared to YOLOX-enabled trackers (e.g., SORT [1], ByteTrack [6]), trackers enabled by CenterNet [9], like CenterTrack [8] and FairMOT [7], show a stronger spatial smoothness on network predictions, especially on center point estimations, as it is explicitly supervised by target-size-related Gaussian kernels [5]. When deploying our method on CenterNet-enabled trackers, each false alarm is shifted by $(\kappa\sigma, \kappa\sigma)$ away from the original detection in different directions, where σ is the radius of the Gaussian kernel corresponding to the original detection. Please note that this slightly differs from attacking YOLOX-enabled trackers (detailed in Sec.3.3.1 in the main text), where we shift each false alarm by $(\kappa w, \kappa h)$.

Table A2: Experiments of attacking CenterTrack with different shift (κ) settings on the MOT17 dataset.

κ	r_{fa}	IDSW$_{\text{im}}$
0.25	0.75	57.92%
0.375	0.86	65.29%
0.5	0.92	74.38%
0.75	0.99	76.80 %
1.0	1.02	78.85 %
1.5	1.07	**79.92**%
2.0	1.10	74.81%
2.5	1.11	68.68%
3.0	1.11	62.72%

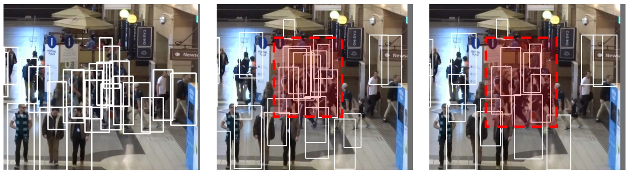
To better demonstrate this spatial smoothness, we list the false alarm ratios r_{fa} under different settings of κ in Table A2. Formally, r_{fa} is calculated by $r_{\text{fa}} = |\tilde{\mathbb{D}}|/|\mathbb{D}|$, where $|\tilde{\mathbb{D}}|$ is the number of false alarms obtained by feeding the detector with the attacked image and $|\mathbb{D}|$ is the expected number of false alarms. According to Table A2, when the spacing between false alarms is not large enough, the false alarm ratio r_{fa} is lower than expected (i.e., $r_{\text{fa}} < 1$). For experiments reported in the main submission, we use $\kappa = 0.5$.

Besides, CenterTrack [8] conducts association based on center representations rather than based on box representations. As a result, it shows less sensitivity to the size prediction of the box. Therefore, no scaling is adopted when attacking CenterTrack (i.e., $s = 1$).

[1] https://github.com/ifzhang/ByteTrack

[2] https://github.com/xingyizhou/CenterTrack

[3] https://github.com/ifzhang/FairMOT



Original detections Expected detections Actual detections

Figure A1: Gaps between expected detections and actual attack results of Hijacking.

B. Weaknesses of One-to-One Design

The Hijacking attacker [3] focuses on misleading the Kalman filter [4] inside the tracker. When deploying it to attack multiple targets simultaneously, each original detection box is translated in the direction opposite to the correct velocity. The attack effectiveness of this one-to-one design (each original box corresponds to one translated box) decreases as the number of targets increases.

For example, in the crowded scene highlighted in Fig. A1, the detection set expected by the Hijacking attack may have an extremely high density. However, due to (1) conflicting perturbation demands from different target individuals and (2) high-density detections being suppressed by NMS, the actual attack results differ significantly from the expected ones. In contrast, we inject multiple false alarms with reasonable density for each original target. Such a one-to-many design is less affected by the above-mentioned factors.

C. Limitation on Attacking Re-Identification-Based Trackers

Table A3: Experiments of attacking FairMOT on MOT17 dataset.

F&F attack	$\mathcal{L}_{\text{emb}}$	β	#iter	#Fm.	ϵ	IDSW$_{\text{im}}$
✓		0.9	60	3	8/255	4.65%
✓		0.5	60	3	8/255	58.25%
✓	✓	0.9	60	3	8/255	51.14%
✓	✓	0.9	60	3	12/255	58.50%

Due to the natural limitation of fooling detectors alone, the effectiveness of the F&F attacker may degrade when attacking some re-identification-based multi-object trackers. This is primarily due to the smooth updating of appearance embeddings inside the tracker, which results in the violation of Eq. 5 and Eq. 6 in the main text. Taking FairMOT [7] as an example, in the new time step t, the appearance embedding of an associated trajectory is updated by

$$\boldsymbol{e}^{t}_{\text{smooth}} = \beta \boldsymbol{e}^{t-1}_{\text{smooth}} + (1-\beta)\boldsymbol{e}^{t}, \tag{1}$$

where β is a smoothness factor and is set to 0.9 by default, indicating the rather low confidence on the new observation $\boldsymbol{e}^t$. As shown in Table A3, by relaxing the β to 0.5, our method achieves a reasonably good attack success rate of 58.25%. In order to promote the attack efficiency under original β settings, we suggest pushing the appearance embeddings of false alarms towards those of original detections by adding one loss item $\lambda_{\text{emb}}\mathcal{L}_{\text{emb}}$ (weighted by $\lambda_{\text{emb}} = 5$) to the targeted loss $\mathcal{L}^{\text{CenterTrack}}_{\text{tgt}}$ (the design for attacking CenterTrack in the main text can be reused to attack FairMOT because these two trackers share the same build-in detector). In specific, we define the loss item as

$$\mathcal{L}_{\text{emb}} = 1 - \cos(\boldsymbol{e}^t, \tilde{\boldsymbol{e}}^t), \tag{2}$$

where $\cos(\cdot,\cdot)$ measures the cosine distance between embeddings, and $\tilde{e}$ indicates the appearance embedding of a false alarm. This leads to the overall targeted loss of attacking FairMOT being designed as

$$\mathcal{L}^{\text{FairMOT}}_{\text{tgt}} = \mathcal{L}^{\text{CenterTrack}}_{\text{tgt}} + \lambda_{\text{emb}}\mathcal{L}_{\text{emb}}. \tag{3}$$

Enabled by attacking the embedding module with the loss item $\mathcal{L}_{\text{emb}}$, the attack success rate reaches 51.14%. Besides, we find that attacks on FairMOT benefit from greater ϵ.

D. Qualitative Analysis

To better illustrate how our method implicitly integrates the association attack, qualitative results are provided in Fig. A2.

ByteTrack [6] and SORT [1] take a probation period of 1 frame. Though they do not spawn new trajectories for false alarms in the first attack frame, our attack has already started to take effect. Specifically, with the original detection being erased, one of the shifted false alarms instead inherits the original identity, misleading trackers to get incorrect estimations (e.g., velocity estimations). This reduces the probability of the original identity being correctly transmitted across the attack. Therefore, our attack method implicitly integrates the association attack without explicitly accessing or forwarding the association component.

The identity switches on isolate targets (highlighted by red triangles) validate the attack example given in Fig. 2 in the main submission. That is, by injecting deceptive false alarms, one of the newly spawned trajectories wins the competition, handing over a wrong identity to the detection after the attack. Besides, deceptive false alarms have a higher efficiency in attacking targets within a crowd (highlighted by yellow triangles) because once a target steals the identity of another target, the latter will also experience an identity switch.

Different from ByteTrack and SORT, no probation period is adopted by CenterTrack [8]. A new trajectory is spawned once it is detected. Besides, CenterTrack employs

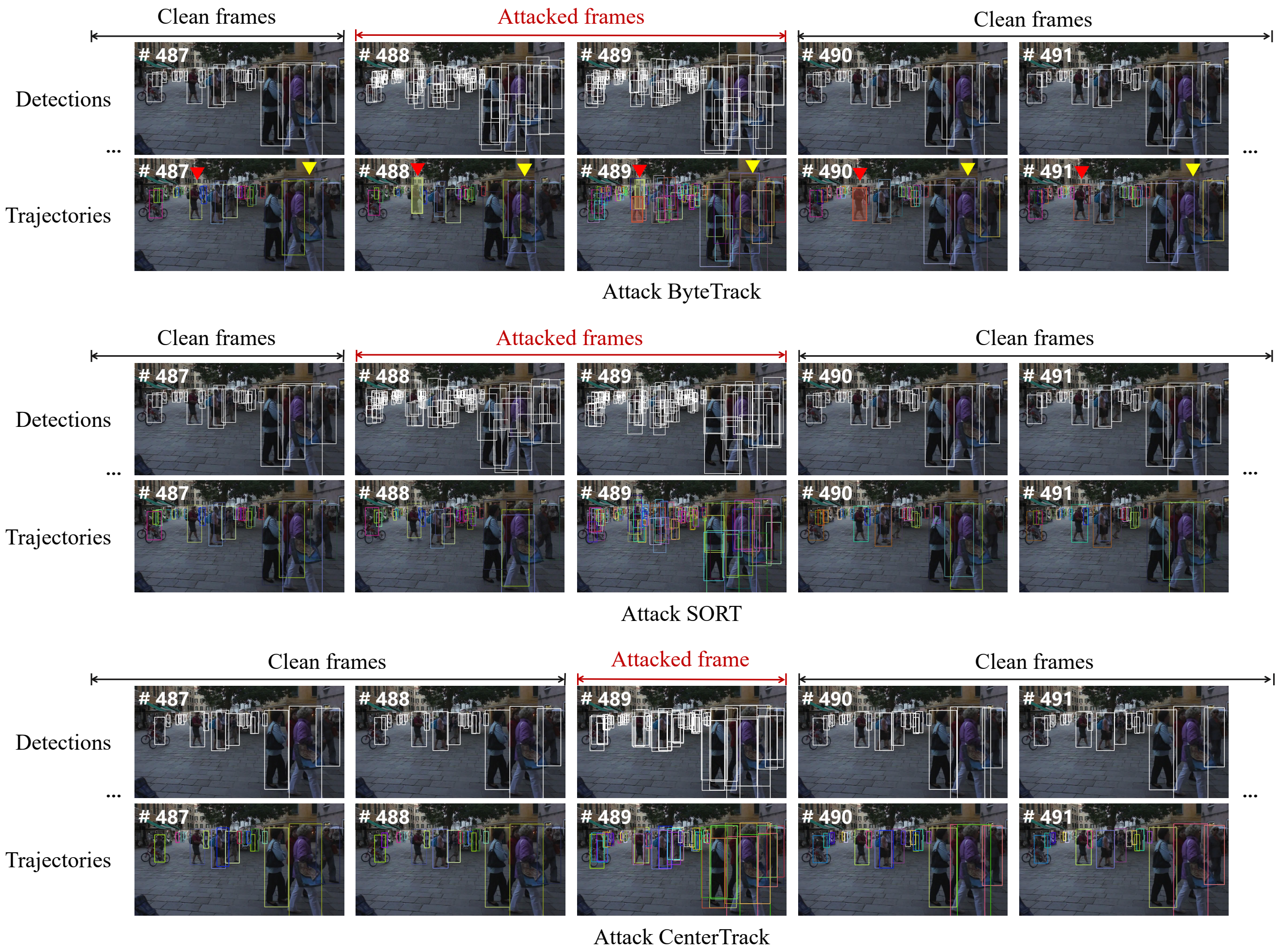


Figure A2: Qualitative results of our method. For each deployment, we list the detection results (i.e., intermediate products) in the first line and the association results (i.e., final outputs of trackers) in the second line. Tracking identities are coded by color. The red triangles show examples of attacking isolated targets and the yellow triangles show examples of attacking targets within a crowd. For more details please refer to the document.

an aggressive trajectory update strategy, placing 100% confidence in the latest observations. This makes CenterTrack more vulnerable to our attacks. By attacking only 1 frame, our method achieves an identity switch rate of about 75%.

References

[1] 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.

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

[3] 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.

[4] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[5] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[6] 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.

[7] 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.

[8] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Track-

ing 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.

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