IJCAI 2025 Poster

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

Woojin Shin1, Donghwa Kang2, Daejin Choi3, Brent Kang2, Jinkyu Lee4, Hyeongboo Baek

1University of Seoul, 2Korea Advanced Institute of Science and Technology(KAIST), 3Incheon National University, 4Sungkyunkwan University

2025.09.04
HyorinJung

Index

Introduction

Detection-Phase Attacks

Initial approaches aim to degrade detection quality by creating false negatives or false alarms

- Lacks Efficiency : Accuracy is only reduced in the specific frames under attack
- No Lasting Impact : The tracker recovers as soon as the attack stops

Disrupting the association phase

Recent methods manipulate object positions to cause Identity (ID) switches during the association phase.

- Lasting Impact
- Lacks Robustness : Can be easily countered by adjusting distance-related parameters
- Predictable: Relies on physically shifting the object's perceived location.

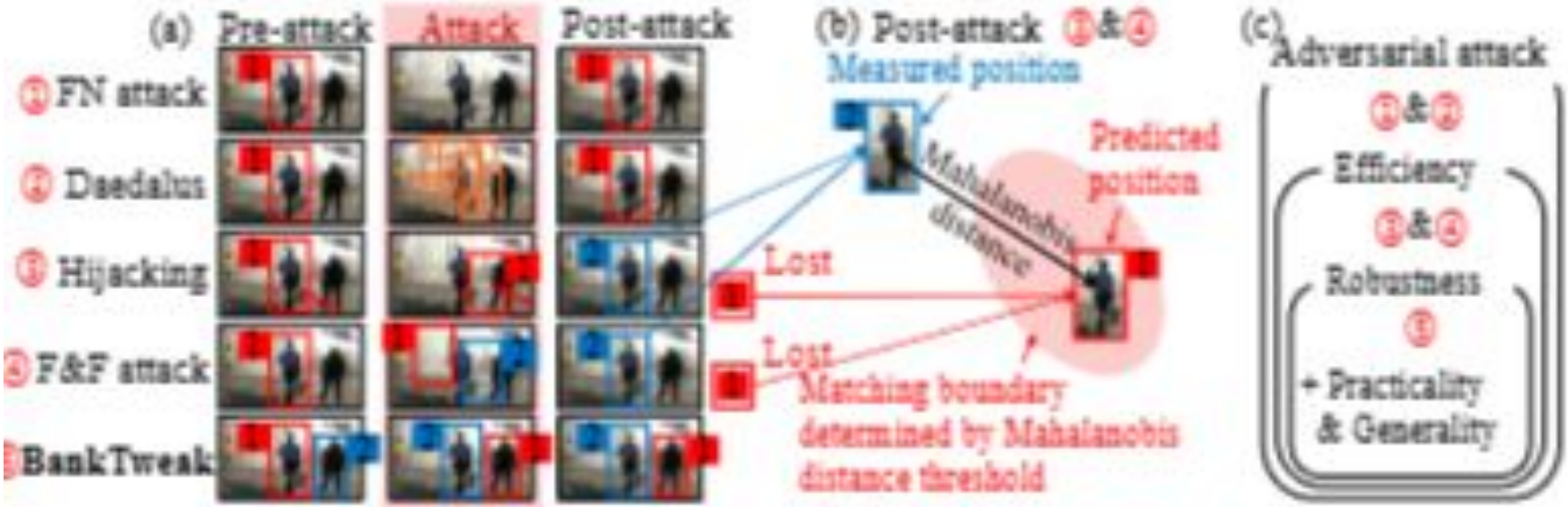
Introduction

Alternative: “Efficiency & Robustness” Solution – BankTweak

- It targets the **feature extractor** in the association phase
- A strategic attack that induces persistent ID switches by manipulating features in the feature bank **without altering object positions**

👉 BankTweak overcomes the limitations of previous methods by targeting the feature bank, the core of the association phase

Introduction



👉 Efficiency & Robustness & Practicality & Generality

Background

Modern Multi-Object Tracking (MOT) works in two main stages to identify objects and follow them across video frames

- **Detention**
  - A detector identifies and draws bounding boxes around all objects of interest within a single frame
- **Association**
  - Links new detections to existing tracks using appearance features and motion cues
  - Sequential matching:
    - Feature-based matching pairs objects with highest feature similarity using the feature bank
    - Motion-based IoU matching handles unmatched objects

Background

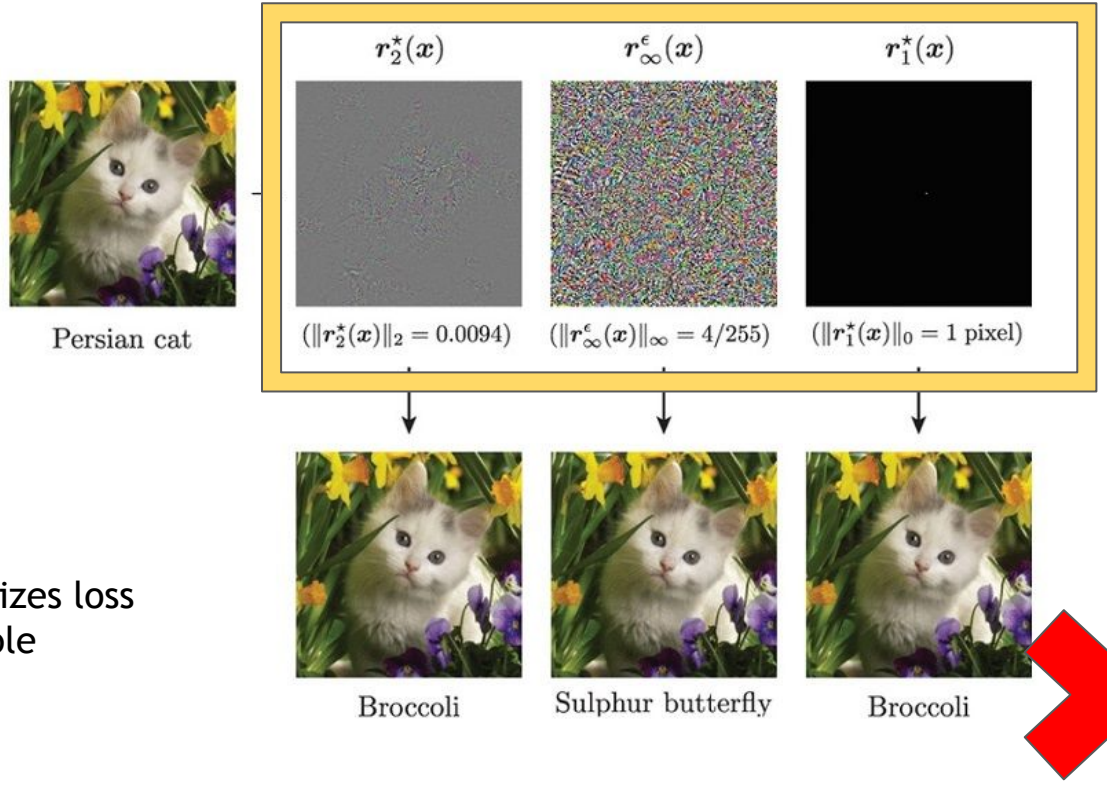
Feature bank

- A storage of appearance features for all tracked objects, used to compare and match new detections with existing tracks across frames
- Maintains historical object representations, enabling the tracker to recognize objects even after short occlusions or frame gaps

Background

Adversarial Perturbation

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



👉 Find perturbation δthat minimizes loss while keeping change imperceptible

Attack formulation

PGD(Projected Gradient Descent)

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

- **α(alpha):** Step size (per-pixel change)
- **ϵ(epsilon):** Max allowed perturbation (ℓ_∞-norm bound)
- **∇(nabla):** Gradient of loss LLL w.r.t perturbation
- **sgn(·):** Uses only the sign (direction) of the gradient

Attack formulation

PGD(Projected Gradient Descent) Steps

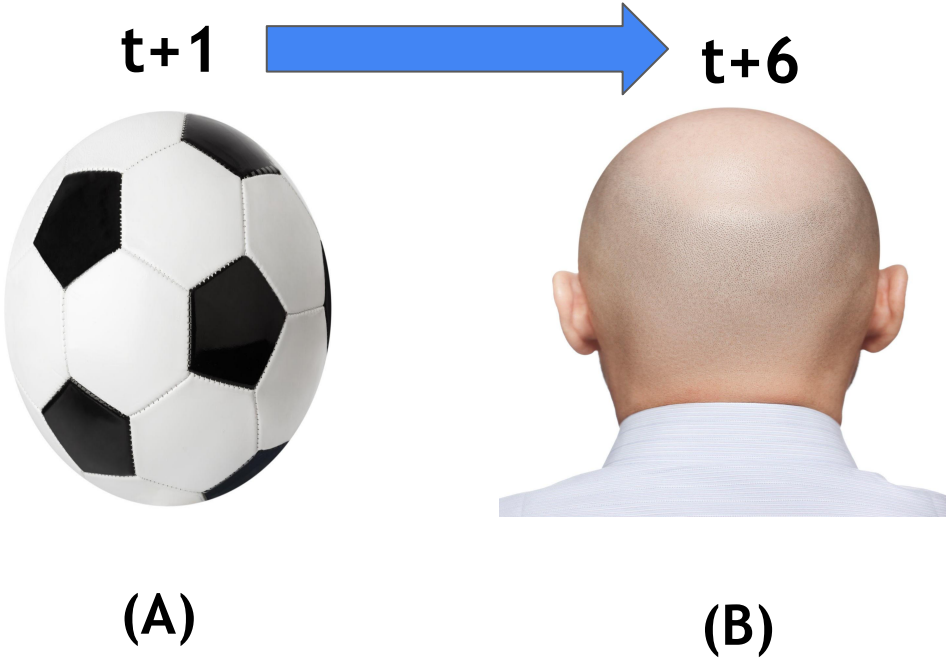
- **Initialization:** $\delta=0$
- **Iterative Update (R steps):**
    - Compute loss L on current perturbed image
    - Compute gradient ∇L
    - Update perturbation: $\delta r+1=\delta r+\alpha \cdot sgn(\nabla L)$
    - **Clipping:**
        - Keep $|\delta| \leq \epsilon$
        - Ensure valid pixel range [0,1]
- **Final Adversarial Image:** $I\sim=I+\delta$

Outcome

- **Imperceptible to humans**
- **Disrupts detector & feature extractor**
- **Effective with only 5 frames**

How BankTweak Works: A Two-Step Process

The attack unfolds over five consecutive frames to induce a **permanent** ID switch between two objects, (A) and (B)



Step 1: The Groundwork (Frames t+1 to t+3)

The goal is to secretly inject carefully crafted features into the feature banks of target objects (A) and (B) without triggering an ID switch prematurely.

Frame t+1	Frame t+2	Frame t+3
Inject 'dummy' features (X|A and Y|B) that are very dissimilar to the original features of A and B.	Inject a feature that looks like B but is generated from A (B|A) into object A's feature bank.	Inject a feature that looks like A but is generated from B (A|B) into object B's feature bank.

Outcome of Step 1

At the end of this phase, the feature banks for both objects are 'poisoned' with misleading information, setting the stage for the final switch.

Step 2: The ID Switch (Frames t+4 & t+5)

With the groundwork laid, BankTweak now exploits the Hungarian matching algorithm, which allocates IDs based on the lowest average feature distance.

Frame t+4: The Switch	**Frame t+5: Solidification**
A specific feature is crafted for object A. The algorithm now finds it 'cheaper' to match the current object A with the history of object B (which contains the poisoned B|A feature). This confusion forces the ID switch.	A similar process is repeated for object B to solidify the swap and prevent it from reverting in the next frame.

Result: A Permanent Switch

From frame t+6 onwards, the objects' original features continue to match with the now-swapped ID histories. The ID switch is permanent, even though no more attacks are being performed

Why Not Just Swap Features Directly?

The Tracker is Too Smart

- Direct attack of injecting B's feature into A's track will fail.
- The feature bank for A still contains many instances of A's true feature. The matching algorithm will correctly prioritize the strong history of true 'A' features, causing any temporary ID switch to revert immediately.

👉 BankTweak's meticulous two-step process is necessary to overcome this resilience.

Solving perturbations : Using Similarity & Dissimilarity Loss

Loss Functions

Similarity Loss	Dissimilarity Loss
$\mathcal{L}^s(\mathbb{F}^*, \mathbb{F}) = \sum \mathcal{C}(F_i^*, F_i),$	$\mathcal{L}^d(\mathbb{F}^*, \mathbb{F}) = -\sum_{F_i^* \in \mathbb{F}^*, F_i \in \mathbb{F}} \mathcal{C}(F_i^*, F_i).$
Make current feature Fi* **similar** to target Fi (minimize cosine distance)	Make current feature Fi* **different** from target Fi (maximize cosine distance)

Solving perturbations : Using Similarity & Dissimilarity Loss

Stepwise Attack

Step 1: Groundwork

Step 2: ID Switch

- Alternate Ls and Ld across frames
- Inject manipulated features into the feature bank
- Exploit Hungarian matching vulnerability
  → achieve permanent ID switch

Experiments : Setting

Efficiency: IDF1, HOTA, ID Switches (IDsw)

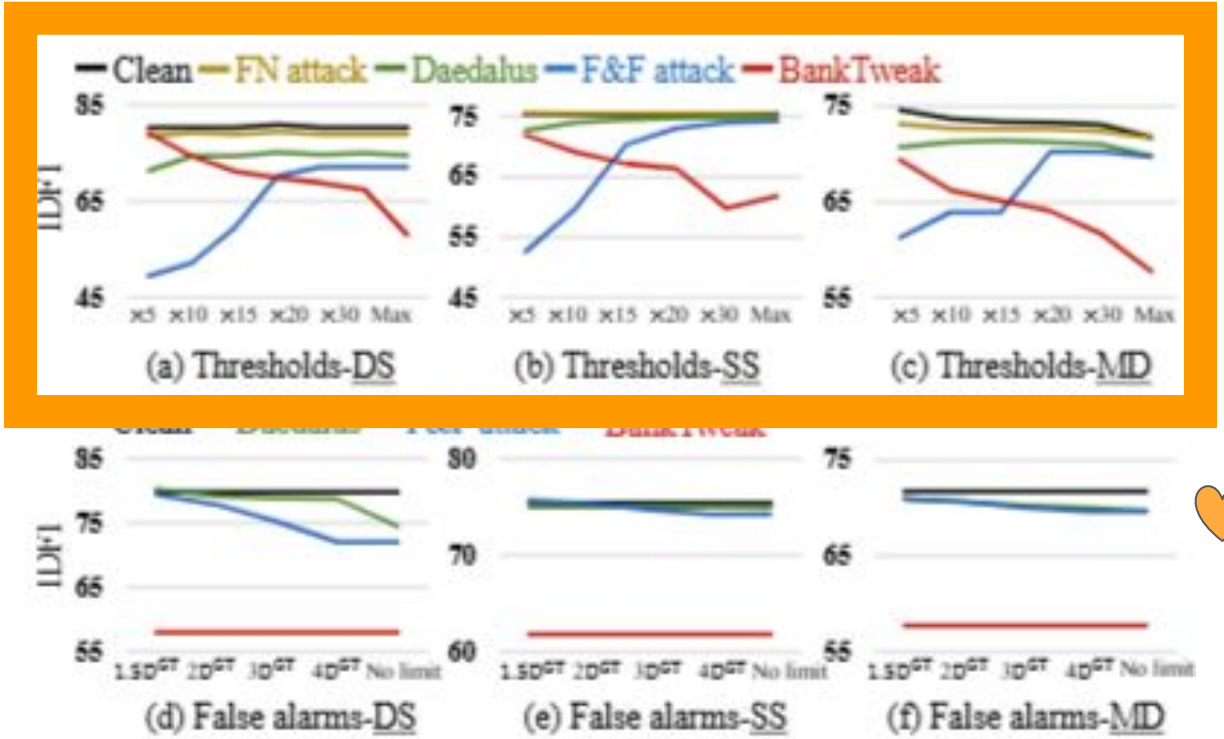
- **HOTA:** $\sqrt{(DetA \times AssA)}$
  - DetA = accurate detection ratio
  - AssA = correctly tracked object ratio
- Accuracy excludes attack frames

Robustness: Measured by varying Mahalanobis distance threshold
Practicality: ρDet & ρID

- ρDet = avg. increase in detections per attack frame / GT
- ρID = increase in ID counts

Generality: No metric; BankTweak designed to be model-agnostic

Experiments : Results



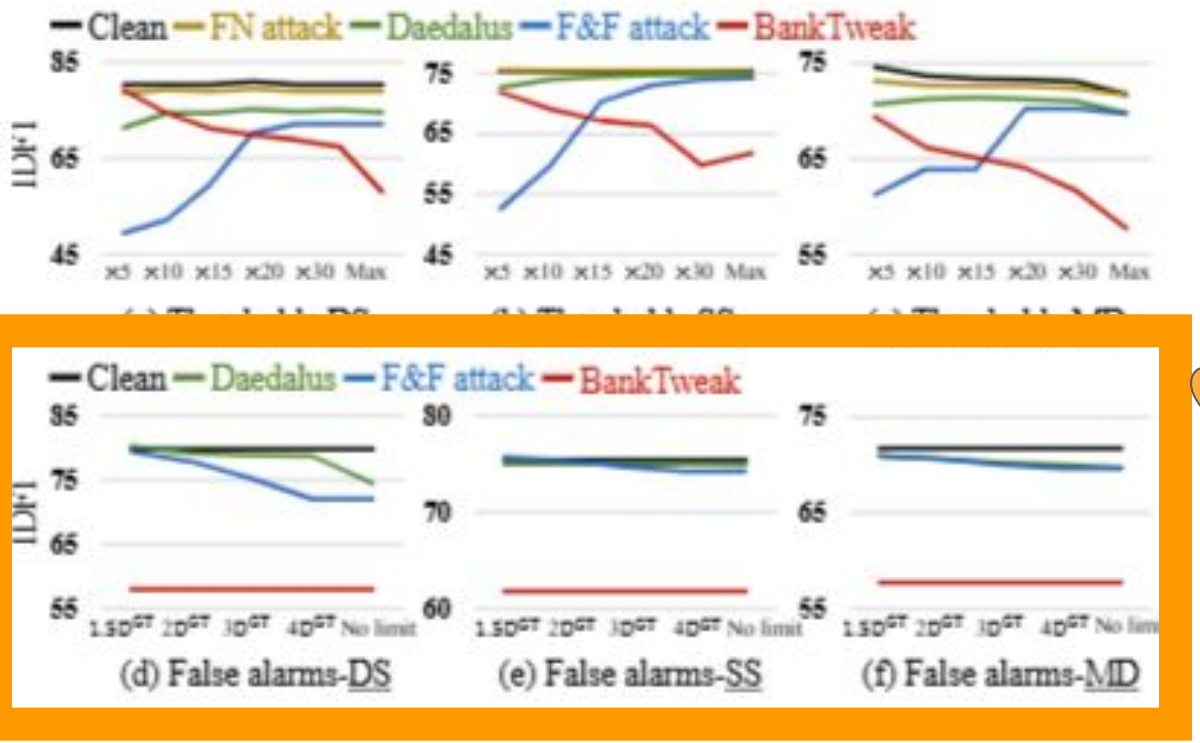
Mahalanobis Threshold

Clean & FN: IDF1 stable, unaffected by threshold

F&F: Attack weakens as threshold increases

BankTweak: Low IDF1, robust to threshold changes

Experiments : Results



Allowed False Positives

Daedalus & F&F: IDF1 drops as allowed false positives increase, but remains higher than BankTweak

Clean & BankTweak: IDF1 stable regardless of false positive limit

Experiments: Ablation study

Mahalanobis Threshold

- F&F, Daedalus: ↑ threshold → ↓ effect
- BankTweak: robust

False Positive Limit

- Clean & BankTweak: stable (no FP)
- Daedalus & F&F: more FP → lower IDF1, still < BankTweak

Step 2

- Removed: 79.78 → 77.52 (small drop)
- Included: 79.78 → 58.01 (large drop)

Iterations (Rs)

- More Rs → stronger attack
- Larger ϵ → fewer Rs needed

Limitation & Future Directions

- The current model is a white-box attack, requiring internal knowledge of the tracker's models
- BankTweak’s transfer-based black-box attacks are feasible when two target objects have **similar features**, allowing ID switches without full model access
- Future work will explore transfer-based black-box attacks, which show initial feasibility.
- The attack requires at least two objects, but this is not a practical limitation in real-world tracking scenarios where scenes are rarely empty.

Conclusion

- BankTweak induces persistent ID switches in multi-object trackers.
- Robust against changes in Mahalanobis distance thresholds.
- Does not generate false positives, ensuring practical applicability.
- Step 2 and iterative updates are crucial for attack efficiency.
- Transfer-based black-box attacks are feasible with similar-feature object pairs.
- Searching similar object pairs is computationally intensive → motivates lightweight models.