RETRACK: LEARNING RECOVERY DECISIONS FROM TRAJECTORY REPLAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual trackers commonly assume limited target displacement across consecutive frames and localize the target within a local search region cropped around the previous prediction. However, the tracker may lose a small target when rapid motion moves it outside this local window or brief occlusion disrupts localization. It may then drift onto background clutter and fail to recover. To recover a lost target, existing trackers combine local tracking with full-frame re-detection, often relying on a separately trained external detector and hand-crafted rules based on current-frame confidence to trigger recovery. Nevertheless, these rules decide whether to switch from current-frame cues, without training on how staying and switching affect the subsequent trajectory. To close this gap, we introduce **ReTrack**, which learns recovery decisions through *trajectory-replay supervision*. By rolling out the stay and switch trajectories over subsequent frames and evaluating both against the ground-truth trajectory, we directly measure the relative advantage of switching. At inference, a lightweight policy predicts this advantage from relative geometric, confidence, and temporal cues. It then decides whether to retain the tracker prediction or switch to a full-frame candidate. These candidates are provided by a full-frame branch that shares the same backbone as the tracker, eliminating the need for a separately trained detector. With full-frame candidates and learned recovery decisions, ReTrack achieves 49.07% AUC on UAV-Anti-UAV, surpassing the previous best result by 5.04% AUC, and improves performance on three additional drone tracking benchmarks.

1 INTRODUCTION

Visual object tracking aims to localize a target throughout a video, given only its box in the first frame. Most visual trackers localize the target within a search region centered on the previous prediction. However, this local search strategy is vulnerable to target loss when tracking small objects that occupy only a few pixels: rapid motion can drive the target outside the search region, while brief occlusion can cause the tracker to drift, as commonly observed in drone tracking from a moving UAV or a ground-based camera (Zhang et al., 2025; Huang et al., 2024a). For instance, on UAV-Anti-UAV, the target is absent from 17% of the frames, and more than half of the sequences have a median target size of at most 30 pixels. Once the target falls outside the local search region, the tracker may lock onto background clutter and then fail to recover it through local search alone (Figure 1(a)).

To address the limitations of local search, existing efforts broadly fall into two paradigms. ***Temporal-memory trackers*** (Zheng et al., 2024; Li et al., 2025a;b) stabilize local matching with historical observations, yet they still search only around the previous prediction and thus cannot recover a target that has already left the search region. ***Tracking-by-detection frameworks*** (Huang et al., 2020; Yan et al., 2019; Dai et al., 2020; Huang et al., 2024b) instead search the full frame with a separately trained re-detector and rely on a hand-crafted confidence rule to decide when the re-detection should replace the local track, as shown in Figure 1(b). In practice, this detector is a separate model that must be trained on the target domain, so the recovery path carries its own training cost. Consequently, the difficulty lies not only in *where* to search but in *when* to switch. Current-frame confidence provides only a local indicator of reliability, failing to capture whether a switch keeps the trajectory stable or triggers a new drift over the following frames.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

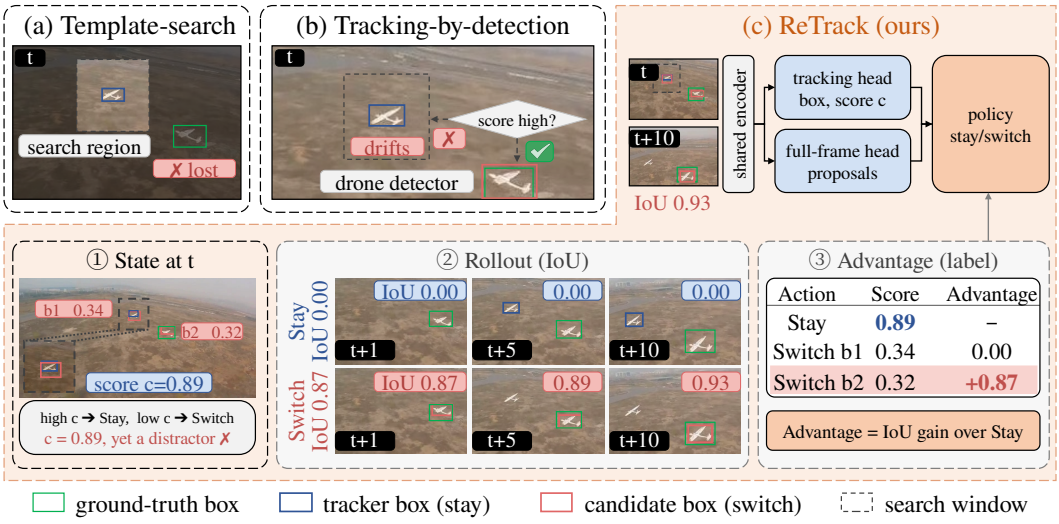


Figure 1: Tracking paradigms. (a) template and local search matching, (b) tracking by detection with a confidence rule, and (c) ReTrack, where a learned policy decides whether to stay with the local track or switch to a full-frame proposal. ①–③ illustrate trajectory replay at one training decision point: staying and switching are rolled out from the same state and compared by IoU.

We observe that the ground-truth trajectories available during training offer a direct way to quantify the effect of a recovery decision over subsequent frames. By comparing the future trajectories of staying versus switching, we can define a ***switching advantage*** that explicitly measures whether a recovery decision improves tracking performance. Specifically, through ***trajectory replay***, we roll out from the same tracker state both the *stay* (local prediction) and *switch* (full-frame candidate) trajectories over subsequent frames. The difference in their mean IoU with the ground truth yields the switching advantage, which serves as a direct supervisory signal for learning recovery decisions. This paired rollout transforms passive bounding-box annotations into counterfactual decision targets, requiring no labeling beyond the existing bounding boxes.

Based on this insight, we propose ReTrack, a tracker that integrates local tracking with a shared full-frame branch and a learned decision policy, as depicted in Figure 1(c). The architecture is designed to decouple proposal generation from decision making. During training, the full-frame head supplies candidate proposals, while the trajectory replay module computes the switching advantage for each candidate by simulating their future trajectories against the ground truth. These advantages supervise a lightweight decision policy, which learns to predict the advantage from relative geometric, confidence, and temporal cues. At inference, the policy operates without access to the future trajectory, relying on these same cues to decide whether to retain the tracker's prediction or switch to a full-frame candidate. To ensure the policy encounters a diverse distribution of target-loss states, we employ out-of-fold cross-fitting: each tracker is trained on a subset of sequences, and decision samples are collected only on the held-out fold. Because each tracker runs on sequences it was not trained on, its failures during collection resemble those at test time.

We evaluate ReTrack on four drone tracking benchmarks. ReTrack achieves the best AUC on three benchmarks, surpassing the previous best by 5.04% AUC on UAV-Anti-UAV, and the recovery pipeline transfers to three additional base trackers. Our contributions are as follows:

- **Trajectory-Replay Supervision.** We replay the stay and switch trajectories from the same state and use their difference as a switching-advantage label that directly supervises recovery decisions.
- **Unified Recovery Framework.** ReTrack adds a full-frame head to the tracker backbone and a lightweight policy that decides whether to switch at each decision point, without a separate detector.
- **Strong Results and Transfer.** ReTrack achieves the best AUC on three drone tracking benchmarks, and the recovery pipeline transfers to three additional base trackers.

2 Related Work

Generic Object Tracking. Siamese trackers (Bertinetto et al., 2016; Li et al., 2019) formulate tracking as template–search matching, while transformer trackers (Chen et al., 2021) replace correlation with attention between template and search-region features. One-stream trackers further merge feature extraction and matching into a shared backbone (Ye et al., 2022; Cui et al., 2022; Wu et al., 2023; Chen et al., 2025), and LoRAT (Lin et al., 2024) adapts a pre-trained backbone (Oquab et al., 2024) with low-rank updates (Hu et al., 2022), while other methods incorporate temporal memory (Zheng et al., 2024; Li et al., 2025a;b). These advances improve local appearance modeling and leverage temporal context, but localization still operates primarily within a search region centered on the previous prediction. Once the target leaves the search region, better local matching alone cannot recover the target, which motivates full-frame re-detection.

Tracking Recovery and Re-Detection. Long-term tracking addresses target loss by complementing local tracking with full-frame search. TLD (Kalal et al., 2012) combines tracking, detection, and verification, while GlobalTrack (Huang et al., 2020), SPLT (Yan et al., 2019), and RTracker (Huang et al., 2024b) perform full-frame re-detection. LTMU (Dai et al., 2020) estimates tracking reliability from present and historical observations, KeepTrack (Mayer et al., 2021) associates candidate objects across frames, and Siam R-CNN (Voigtlaender et al., 2020) resolves long-term trajectories with full-frame proposals and offline dynamic programming. A common design in this line pairs a detector with a rule that decides when the local track should be replaced. Full-frame search thus answers where to recover, but when to switch is still decided largely by current-frame confidence or hand-crafted rules. Segmentation-based trackers built on SAM 2 (Ravi et al., 2025), such as SAMURAI and DAM4SAM (Yang et al., 2026; Videnovic et al., 2025), propagate masks and memory across frames, but they run a segmentation foundation model on every full frame. ReTrack instead keeps a box tracker, evaluates its full-frame branch only at decision points, and learns when a proposal should replace the local track.

Learning Recovery Decisions. Several methods replace fixed rules with learned decisions. ADNet (Yun et al., 2017) learns discrete tracking actions with reinforcement learning, Supančić and Ramanan (Supancic III & Ramanan, 2017) learn policies over tracking, re-initialization, and model update from sampled returns. Sequence-level training (Kim et al., 2022) and RELO (Chen et al., 2026a) optimize the tracker's localization with sequence-level rewards, while ODSTrack (Chen et al., 2026b) learns a value function for switching to a separate detector. These methods supervise the current tracking state or the selected action, rather than directly comparing staying and switching from the same state. ReTrack instead replays both trajectories through subsequent frames and uses their mean-IoU difference as the recovery target. Following DAgger (Ross et al., 2011), iterative collection gathers decision samples from states encountered under the current policy, while cross-fitting prevents these samples from being dominated by sequences used in tracker training.

3 Method

As depicted in Figure 2, ReTrack integrates local tracking with a shared full-frame branch and a learned decision policy. A unified backbone encodes the template and the local search region into a shared representation, with the motion residual added to the input tokens, and, at each ***decision point***, a frame at which recovery is evaluated, also encodes the full frame. The tracking head localizes the target within the search region, while the full-frame head generates K recovery proposals from the entire frame. At a decision point, the decision policy predicts the switching advantage of every proposal over staying with the current track. Counterfactual trajectory replay supplies the switching-advantage labels used to train this policy. We describe the architecture in Section 3.1, trajectory-replay supervision in Section 3.2, and the decision policy, training, and inference in Sections 3.3 and 3.4.

3.1 ReTrack Architecture

Base Tracker. We build on LoRAT, which adapts a frozen pre-trained backbone (Oquab et al., 2024) with low-rank updates (Hu et al., 2022). The template and the search region cropped around the previous prediction are patchified into tokens $(\boldsymbol{z}_0, \boldsymbol{s}_0)$, then processed jointly by L transformer blocks. The tracking head consists of two three-layer MLPs for score prediction and box regression. For each search token, the tracking head predicts a bounding box and a score. At inference, the

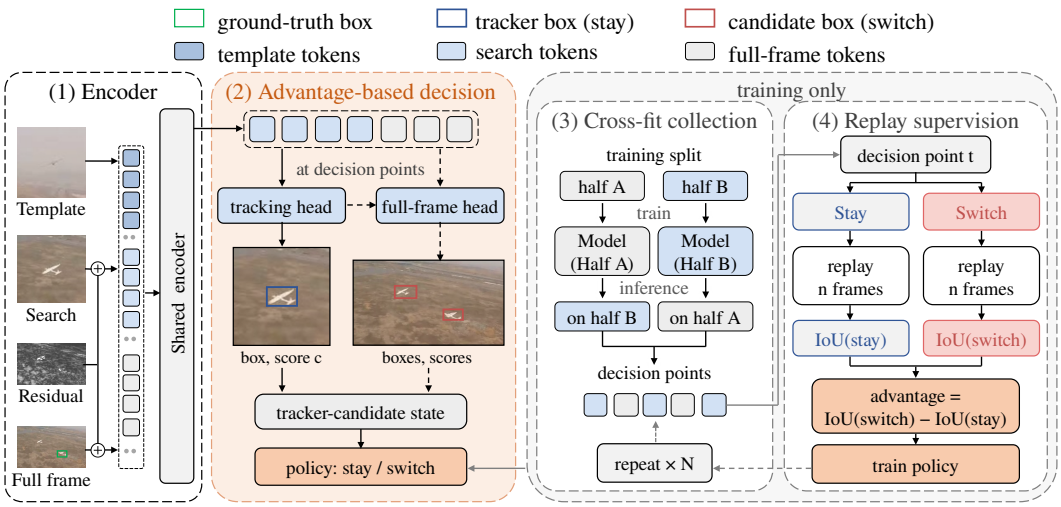


Figure 2: Overview of ReTrack. (1) A shared encoder processes the template, the local search region and, at decision points, the full frame, with the projected motion residual added to the input tokens. (2) The tracking head predicts the current box with confidence c, the full-frame head supplies K candidate boxes, and at a decision point the policy chooses between staying and switching. (3) Trackers trained on one half of the split collect decision points on the other half, repeated over N rounds. (4) Replay rolls out both decisions for n frames, and their IoU difference is the advantage label. Panels (3) and (4) run only during training. $\oplus$ denotes element-wise addition of the projected residual tokens.

box with the maximum score is selected as the tracker output, and its score c denotes the tracking confidence. For small targets, the base tracker may have limited appearance evidence. We therefore derive a motion residual from the current grayscale frame I_t and an aligned earlier frame $I_{t-\Delta}$ as an additional motion cue. The motion residual is $R_t = |I_t - \mathcal{W}(I_{t-\Delta})|$, where $\mathcal{W}$ aligns the preceding frame $I_{t-\Delta}$ to I_t by phase correlation and Δ is the temporal offset. The residual is projected by a zero-initialized patch embedding and added to the input tokens of both branches.

Full-Frame Branch. This branch searches the entire frame to propose candidate targets, including locations outside the local search region. The frame is resized to $W_f \times H_f$ with aspect-ratio-preserving padding, patchified, and projected by the shared patch embedding into tokens $\boldsymbol{f}_0$, augmented with interpolated positional embeddings and a learned branch identifier. The resolution $W_f \times H_f$ is chosen to follow the aspect ratio of the frame while keeping the token count of this branch below that of the local search region, which bounds the cost of running it at decision points. These tokens pass through the shared L transformer blocks independently of the tracking tokens, feeding into a full-frame head with the same structure as the tracking head but separate weights. The K highest-scoring predictions serve as recovery proposals for the decision policy.

3.2 Trajectory-Replay Supervision

Switching Advantage. At a decision point, the tracking head holds a box b^{trk} with confidence c inside its search region, and the full-frame head supplies candidates $\{b_1, \ldots, b_K\}$ with scores $\{c_1, \ldots, c_K\}$. Re-centering the local search region at a box b and tracking forward for n frames defines a ***trajectory rollout***. The template and all tracker parameters stay fixed during the rollout, so the stay and switch rollouts differ only in where tracking resumes. Let $\rho(b, n)$ denote the mean intersection-over-union (IoU) between this rollout and the ground-truth trajectory. The advantage of switching to candidate k is defined by the difference in rollout quality:

$$a_k \;=\; \rho(b_k, n) \;-\; \rho(b^{\text{trk}}, n), \tag{1}$$

Thus, $a_k > 0$ indicates that switching to candidate k yields a better trajectory than staying.

Paired Rollouts. Trajectory replay evaluates both decisions from the same tracker state (Figure 2(4)). At a training decision point, we roll out the stay trajectory for n frames from b^{trk} and each switch trajectory from its candidate b_k, with all rollouts sharing the template, backbone, and tracker parame-

ters at that point. The mean per-frame IoU of each trajectory forms $\rho(\cdot, n)$ in Equation (1), directly giving the switching advantage a_k. These rollouts serve solely to construct advantage labels offline.

Decision Sample Collection. Before any policy is learned, decision samples can only be collected by switching with the hand-crafted rule, yet a policy trained only on such samples would encounter states at inference that are absent from its training data. Following DAgger (Ross et al., 2011), the collector runs N rounds, using the rule in the first round and the current policy in later rounds, and samples from all rounds are pooled to train the decision policy. We further avoid collecting on sequences used to train the same tracker: the training split is partitioned into two folds, a separate tracker is trained on each fold, and each tracker collects decision samples only from its held-out fold. This cross-fitted collection exposes the policy to target-loss states on unseen sequences. These fold-specific trackers are used only for offline collection, while the final tracker evaluated in all of our experiments is trained on the complete UAV-Anti-UAV training split.

3.3 Online Decision Policy

Relative Decision Representation. Given the limited appearance information of small targets, the policy estimates the switching advantage a_k (Equation (1)) of each recovery candidate b_k from cues that compare the candidate with the current track. Tracker reliability is captured by the current confidence and its temporally smoothed value. Candidate quality is captured by the proposal score and its ranking among the K proposals. The spatial relation between the two boxes is encoded by their normalized center distance and scale ratio. The temporal context is encoded by the time elapsed since the previous switch. These quantities are concatenated into a relative feature vector $\boldsymbol{x}_k$. The representation is appearance-free. It relies on geometric, confidence, and temporal cues alone, excluding image texture, target category, and the absolute position in the image.

The resulting feature vector is mapped by a three-layer MLP $\hat{a}_\theta$ with parameters θ to the predicted switching advantage. At a decision point, the policy evaluates the recovery proposals independently and ranks them by $\hat{a}_\theta(\boldsymbol{x}_k)$. Therefore, the policy learns whether switching improves on staying, rather than estimating whether the tracker is still reliable at the current frame.

3.4 Training and Inference

Training Objective. Both heads are supervised at the token cells covered by the ground-truth box, with a binary cross-entropy on the score map and a GIoU loss (Rezatofighi et al., 2019) on the predicted boxes. The tracker backbone is optimized end-to-end via a multi-task objective combining both task branches:

$$\mathcal{L} = \mathcal{L}^{\text{trk}}_{\text{BCE}} + \mathcal{L}^{\text{trk}}_{\text{GIoU}} + \mathcal{L}^{\text{det}}_{\text{BCE}} + \mathcal{L}^{\text{det}}_{\text{GIoU}}, \tag{2}$$

where the superscripts denote the tracking and full-frame branches, respectively. With the tracker frozen, let $\mathcal{D} = \{(\boldsymbol{x}_k, a_k)\}$ denote the collected decision samples, each pairing a relative feature vector with its switching advantage. The policy is fitted by

$$\theta^* = \arg\min_{\theta} \sum_{(\boldsymbol{x}_k,\, a_k)\in\mathcal{D}} H_\delta\big(\hat{a}_\theta(\boldsymbol{x}_k) - a_k\big), \tag{3}$$

where H_δ is the Huber loss with δ=0.2, which limits the influence of large prediction errors.

Inference. The full-frame head is activated at decision points, which are triggered when the tracking confidence falls below a fixed level or every M frames (Section A). At each decision point, the tracker selects the candidate with the largest predicted advantage. Let τ be the decision threshold, t_{last} the last switch frame, and Δ_{cool} the minimum interval between two switches; the output is

$$k^* = \arg\max_k \hat{a}_\theta(\boldsymbol{x}_k), \qquad b_t^{\text{out}} = \begin{cases} b_{k^*} & \text{if } \hat{a}_\theta(\boldsymbol{x}_{k^*}) > \tau \text{ and } t - t_{\text{last}} \geq \Delta_{\text{cool}}, \\ b^{\text{trk}} & \text{otherwise.} \end{cases} \tag{4}$$

The selected box centers the local search region at frame $t+1$.

4 EXPERIMENTS

4.1 IMPLEMENTATION DETAILS

Training. We adopt a ViT-B/14 (Dosovitskiy et al., 2021) backbone initialized with DINOv2 (Oquab et al., 2024) pre-trained weights and adapted with low-rank adapters (Hu et al., 2022) of rank $r = 64$. The template and the local search region cover 2^2 and 4^2 times the target box area and are resized to 112×112 and 224×224, respectively. Both tracking and full-frame branches are optimized end-to-end for 45 epochs on two NVIDIA RTX 5090 GPUs, sampling uniformly across generic tracking benchmarks (LaSOT (Fan et al., 2019), GOT-10k (Huang et al., 2021), TrackingNet (Müller et al., 2018), COCO (Lin et al., 2014), VastTrack (Peng et al., 2024)) and the UAV-Anti-UAV training split. The full frame is resized with aspect-ratio-preserving padding to 280×168, which approximately matches the 16:9 aspect ratio of most UAV-Anti-UAV frames. The resulting 20×12 token grid contains only 240 tokens, fewer than the 256 tokens of the local search region. The decision policy, a three-layer MLP with a hidden dimension of 64, is trained on decision samples gathered over $N{=}3$ collection rounds with a rollout horizon of $n{=}10$ frames, minimizing a Huber loss.

Inference and Evaluation. At inference, we use $K{=}3$ recovery proposals, a decision threshold $\tau{=}0.05$, and a cooldown $\Delta_{\text{cool}}{=}10$ frames, with further details in Section A. We evaluate ReTrack on UAV-Anti-UAV (Zhang et al., 2025), Anti-UAV300 (Jiang et al., 2023), Anti-UAV410 (Huang et al., 2024a), and DUT-Anti-UAV (Zhao et al., 2022) and report Success (AUC), Precision (P), and Normalized Precision (P_{norm}). Table 1 compares ReTrack with recent generic trackers and two additional baselines: MambaSTS, released with UAV-Anti-UAV, and LoRAT-ID, which retrains LoRAT with the same training data as ReTrack. All generic trackers use their publicly released checkpoints and are scored under the same evaluation protocol as ReTrack on all four benchmarks.

4.2 COMPARISON WITH STATE-OF-THE-ART TRACKERS

Table 1: Comparison with state-of-the-art trackers on four anti-UAV benchmarks. Best and second-best results are shown in **bold** and underline.

Method	Venue	UAV-Anti-UAV			Anti-UAV300			Anti-UAV410			DUT-Anti-UAV		
		AUC	P	P_{n}	AUC	P	P_{n}	AUC	P	P_{n}	AUC	P	P_{n}
Generic training													
KeepTrack	ICCV2021	31.39	44.93	35.23	65.65	92.09	77.18	55.50	77.43	62.76	63.16	85.34	74.04
OSTrack	ECCV2022	29.63	39.38	31.92	58.29	77.02	65.59	42.96	57.69	47.27	62.51	83.31	70.90
DropTrack	CVPR2023	37.95	51.57	41.14	66.47	88.92	75.45	52.64	71.53	58.51	66.00	85.96	73.76
GRM	CVPR2023	29.67	38.90	31.48	55.73	73.24	62.48	42.86	57.19	46.84	62.50	83.79	71.62
CiteTracker	ICCV2023	29.80	39.03	31.59	56.80	74.92	64.04	41.80	55.76	45.55	63.53	83.66	71.95
ROMTrack	ICCV2023	30.60	40.76	32.33	60.82	80.84	68.68	47.32	64.32	52.92	59.26	78.00	66.86
LoRAT	ECCV2024	37.78	50.97	41.18	64.02	87.47	74.43	54.05	75.29	60.72	60.63	81.38	68.37
ODTrack	AAAI2024	31.95	42.05	33.93	53.70	74.22	63.02	44.03	62.06	51.03	57.30	77.56	66.88
AQATrack	CVPR2024	36.31	48.39	38.31	66.23	88.93	75.79	53.48	72.59	59.72	65.70	86.74	74.04
MambaLCT	AAAI2025	38.88	52.14	41.52	–	–	–	–	–	–	–	–	–
DUTrack	CVPR2025	37.63	51.22	40.85	55.91	79.90	68.57	48.58	68.55	56.71	63.90	87.55	75.10
SGLATrack	CVPR2025	24.74	32.65	25.82	51.63	67.08	57.29	36.35	48.02	39.15	54.61	73.34	62.27
LETrack	CVPR2026	28.87	39.14	30.87	61.73	82.78	70.65	49.99	67.88	55.90	61.48	83.08	70.54
UTPTrack	CVPR2026	27.18	35.11	28.46	55.09	72.93	62.51	41.09	55.26	45.34	63.52	83.79	71.75
UETrack	CVPR2026	32.17	43.16	34.09	65.82	86.33	73.40	53.61	71.00	59.31	62.59	79.61	70.67
RELO	ICML2026	38.33	51.19	40.56	65.75	86.06	73.61	51.95	69.55	57.12	**67.23**	85.91	74.85
Benchmark-specific training													
MambaSTS	arXiv2025	44.03	61.44	49.57	–	–	–	–	–	–	–	–	–
LoRAT-ID	–	42.22	56.15	46.44	64.63	86.91	74.76	54.50	75.37	61.69	62.68	83.13	71.50
ReTrack	–	**49.07**	**65.84**	**54.17**	**69.94**	**94.67**	**81.49**	**61.75**	**85.56**	**69.87**	66.91	**88.07**	**75.63**

UAV-Anti-UAV. UAV-Anti-UAV contains 420 test sequences, in which the target is absent from 17% of the frames and more than half of the sequences have a median target size of at most 30 pixels. ReTrack achieves 49.07% AUC, outperforming the previous best tracker MambaSTS by 5.04%. Across the seven attributes in Figure 3, the largest gains over LoRAT-ID appear under fast motion and out-of-view, by 6.75% and 6.08% AUC, where the target is most likely to leave the search region, while full occlusion gains the least, since no proposal can relocate an absent target.

Anti-UAV300. Anti-UAV300 provides 100 thermal infrared test sequences with relatively large targets, as only 12% of them have a median target size of at most 30 pixels. ReTrack achieves 69.94% AUC, surpassing recent UETrack by 4.12%.

Anti-UAV410. Anti-UAV410 is a thermal infrared benchmark whose 120-sequence test split contains smaller targets than Anti-UAV300, with 33% of its sequences at a median size of at most 30 pixels. ReTrack achieves 61.75% AUC, surpassing UETrack by 8.14%. This margin is larger than on Anti-UAV300, whose targets are mostly larger than 30 pixels.

DUT-Anti-UAV. DUT-Anti-UAV is a small benchmark whose tracking subset contains 20 visible-light sequences at 1920×1080 resolution, and every sequence has a median target size above 30 pixels. ReTrack achieves the best precision and normalized precision, while its AUC is slightly below RELO. The small gap reflects the absence of small targets.

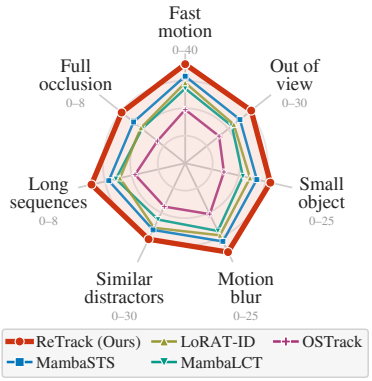


Figure 3: Attribute-wise success on UAV-Anti-UAV.

4.3 Ablation Studies

We conduct ablation studies on UAV-Anti-UAV to analyze the recovery architecture, decision policy, and replay supervision. Shaded rows denote the default configuration.

Recovery Components. As shown in Table 2, ReTrack improves the in-domain base tracker, and removing any recovery component reduces performance. We first retrain LoRAT on the UAV-Anti-UAV training split in (b), which improves AUC from 37.78% to 42.22%, and then remove each component of ReTrack in turn. Disabling switching in (d), so that the full-frame head is retained but never invoked at inference, reduces AUC by 7.86%. Replacing the learned decision in (e) with the hand-crafted rule, which switches to the top proposal when the tracker confidence falls below 0.3, lowers AUC by 3.10%. Removing the motion residual in (f), the residual alignment in (g), or out-of-fold collection in (h) each lowers AUC by 0.6% to 1.2%. Rows (d) and (e) show that switching to full-frame proposals improves performance over never switching, while the learned decision further improves on the hand-crafted rule. Figure 4(a) shows that the gain over the base tracker grows as the target becomes smaller, from 2.5% AUC above 60 pixels to 9.1% at 15 pixels or below.

Table 2: Component ablation.

#	Setting	AUC	P	P_{n}	Δ
(a)	LoRAT	37.78	50.97	41.18	–
(b)	+ in-domain training	42.22	56.15	46.44	–
(c)	ReTrack	49.07	65.84	54.17	–
(d)	− switching	41.21	55.37	45.51	-7.86
(e)	− learned decision	45.97	61.22	50.69	-3.10
(f)	− motion residual	47.89	63.12	52.11	-1.18
(g)	− residual alignment	48.50	64.92	53.49	-0.57
(h)	− cross-fitted collection	48.28	64.46	53.20	-0.79

Stay-or-Switch Decision. To verify the learned decision, we fix the tracker, proposals, and decision points and vary only the switching strategy in Table 3. Random switching in (b), at a rate matched to the learned policy, and always switching to the top-ranked proposal in (c) both fall below never switching in (a), while inverting the predicted advantage in (d) collapses AUC to 27.50%, showing that the policy also identifies harmful switches. The learned advantage in (e) exceeds random switching by 8.28%, so the gain comes from selecting the rare beneficial switches rather than from switching more often. As Figure 4(b) shows, AUC varies by only 0.34% as τ ranges from 0.05 to 0.30, and large targets benefit from a higher threshold, which avoids unnecessary switches because the local tracker rarely loses them.

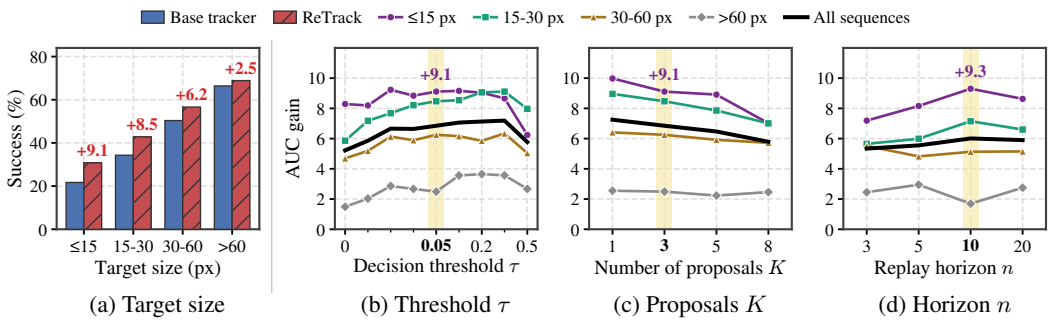
Table 3: Recovery decision variants.

#	Setting	AUC	P	P_{n}
(a)	Never switch	41.21	55.37	45.51
(b)	Random switch	40.79	53.82	44.87
(c)	Always switch	39.27	52.03	43.12
(d)	Inverted advantage	27.50	35.54	30.13
(e)	Learned advantage	49.07	65.84	54.17

Proposal Source. To assess whether recovery depends on the decision rather than an additionally trained detector, we evaluate the hand-crafted rule and the learned decision with proposals from a drone-specific detector and the shared head. As shown in Table 4, the drone-

Table 4: Proposal source and recovery decision.

Proposals	Hand-crafted rule			Learned decision		
	AUC	P	P_{n}	AUC	P	P_{n}
Drone-specific detector	49.75	66.25	54.74	48.93	65.71	54.00
Shared head	45.97	61.22	50.69	49.07	65.84	54.17



(a) Target size (b) Threshold τ (c) Proposals K (d) Horizon n

Figure 4: Analysis on UAV-Anti-UAV. (a) Success of the base tracker and ReTrack by target size. (b)–(d) AUC gain of ReTrack over the base tracker for each target size under different decision thresholds τ, numbers of proposals K, and replay horizons n. Shaded columns mark the default setting, and the number in each column is the gain on targets of at most 15 pixels.

specific detector with the hand-crafted rule reaches the highest AUC, 49.75%, but it requires a separate model trained on the target domain at 960-pixel resolution, whereas ReTrack comes within 0.68% with the shared head alone and slightly exceeds the same policy with detector proposals. The rule accepts the top proposal once the tracker confidence is low, so its accuracy follows the quality of that proposal, whereas the learned decision compares each proposal with the current track. The number of proposals K leads to the same conclusion. Increasing K from 1 to 8 in Figure 4(c) lowers the gain on targets of at most 15 pixels from 9.98% to 7.02% AUC, whereas the gain above 60 pixels barely changes. Therefore, the extra lower-scoring proposals act mainly as distractors for tiny targets, and recovery depends on selecting the correct proposal rather than adding more.

Policy Features. As shown in Table 5, we ablate tracker confidence, candidate score, and geometry to assess their contributions to the relative decision representation. Tracker confidence is the most important input, and removing it in (b) lowers AUC by 5.90%. Removing the candidate score in (c) or geometry in (d), which describes the center distance and scale ratio between a proposal and the current tracker output, also reduces AUC, and partial feature sets in (e) and (f) recover only part of the gap. Together, these results show that recovery decisions cannot rely on tracker confidence or proposal scores alone, but require comparing each candidate with the current track.

Table 5: Policy input and replay target.

#	Setting	AUC	P	P_{n}
(a)	ReTrack	49.07	65.84	54.17
Policy input				
(b)	w/o tracker conf.	43.17	57.95	47.65
(c)	w/o candidate score	47.59	63.97	52.46
(d)	w/o geometry	46.69	62.82	51.56
(e)	Confidence only	47.61	63.99	52.56
(f)	Two scores only	47.42	63.44	52.25
Replay target				
(g)	Binary advantage	49.08	65.91	54.15
(h)	w/o stay rollout	47.01	64.29	51.90

Replay Supervision. To assess trajectory replay supervision, we evaluate its target and rollout horizon. As shown in Table 5, the binary advantage in (g) matches the full switching advantage in (a), whereas the switch-rollout IoU in (h) lowers AUC by 2.06%. The IoU of the switch trajectory alone cannot determine whether switching improves tracking. Together, these results show that replaying both the switch and stay trajectories provides the necessary supervision, while a binary target is sufficient for the recovery decision. We next assess how the rollout horizon affects the trajectory comparison. As shown in Figure 4(d), with one collection round, extending the horizon from n=3 to n=10 increases the gain on tiny targets from 7.19% to 9.30%, whereas medium and large targets, which are rarely lost, benefit little from a longer horizon.

4.4 Additional Analyses

Qualitative Analysis. We visualize three recovery sequences and one failure case in Figure 5. In rows 1–3, both methods lose the target after local tracking fails, and LoRAT then drifts onto background clutter. ReTrack instead switches to a full-frame proposal with a positive predicted advantage and resumes tracking the target. In row 4, ReTrack switches to a visually similar distractor with a high proposal score. Because the decision representation excludes appearance, the policy cannot separate a distractor with similar scores and geometry from the target.

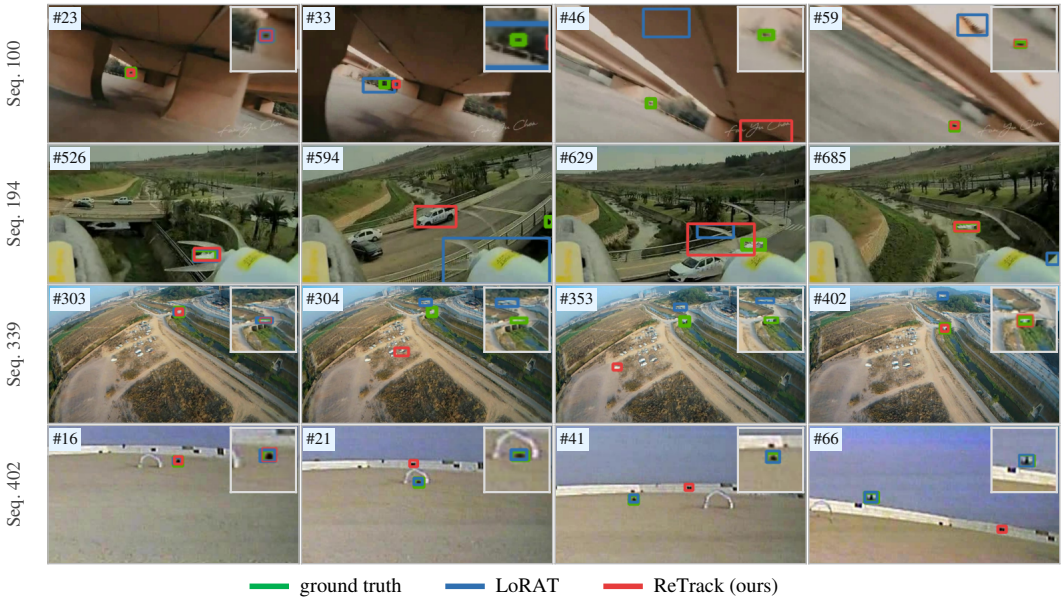


Figure 5: Qualitative comparison with LoRAT. Rows 1–3 show recoveries after target loss, and row 4 shows a failure on a visually similar distractor.

Cross-Tracker Transfer. To assess the transferability of the recovery pipeline, we integrate it into OSTrack, LETrack, and UTPTrack. As shown in Table 6, ReTrack improves AUC over every base tracker, by 2.19% to 6.85%. These three trackers obtain their confidence from a Gaussian heatmap peak rather than the IoU-calibrated classification score of LoRAT. Therefore, the recovery pipeline transfers across architectures and confidence supervision. The gain is largest when the tracker confidence reflects localization quality, as on LoRAT (Section D).

Table 6: Transfer across base trackers.

Tracker	Base			+ ReTrack		
	AUC	P	P_{n}	AUC	P	P_{n}
LoRAT	42.22	56.15	46.44	**49.07**	**65.84**	**54.17**
OSTrack	37.44	50.95	41.25	**43.12**	**59.14**	**48.00**
LETrack	34.89	48.99	38.69	**37.08**	**51.62**	**41.36**
UTPTrack	35.40	48.18	38.72	**40.73**	**56.34**	**45.56**

Computational Cost. The recovery components add little cost to the base tracker. As shown in Table 7, the full-frame head adds 2.71M trainable parameters. Because it runs only at decision points, ReTrack requires 22% more MACs per frame on average than LoRAT-ID. The policy adds fewer than 5K parameters and negligible computation.

Table 7: Computational cost.

Model	AUC	Trainable Params (M)	MACs (G)
LoRAT-ID	42.22	12.98	27.7
ReTrack	49.07	15.69	33.7

5 Conclusions

In this work, we present ReTrack, which combines a full-frame head with a policy learned from trajectory replay to make recovery decisions. During training, trajectory replay rolls out the stay and switch trajectories from the same state, and their mean-IoU difference defines the switching advantage that supervises the policy. The full-frame head answers where to recover, while the learned policy decides when to switch. ReTrack achieves the best AUC on three of the four benchmarks, and the recovery pipeline transfers to three additional base trackers. Recovery remains bounded by the proposals, decision points are still triggered by a fixed confidence level and period, and because the decision representation excludes appearance, the policy cannot reject a visually similar distractor with a high proposal score. Future work can add appearance cues to the decision, strengthen proposal generation, and learn when to trigger decision points.

Reproducibility Statement

Section 4.1 summarizes the architecture and training, while Section A covers the collection and inference. The UAV-Anti-UAV results come from its official toolkit, the other benchmarks are scored with the same success and precision definitions, with success computed over 21 IoU thresholds, and the raw prediction files of all reported results will be released together with the code.

AI Use Statement

In this work, we used generative AI coding assistants to help implement the evaluation and experiment code and to check the consistency of reported numbers. Additionally, we used generative AI tools to edit the manuscript to improve readability and to search for and identify relevant literature. We have reviewed all AI-assisted work and verified the code against the raw result files. We take responsibility for the final content of this work, including text, claims, or artifacts produced with AI assistance.

References

Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, 2016.

Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. In *ICCV*, 2023.

Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021.

Xin Chen, Ben Kang, Wanting Geng, Jiawen Zhu, Yi Liu, Dong Wang, and Huchuan Lu. SUTrack: Towards simple and unified single object tracking. In *AAAI*, 2025.

Xin Chen, Chuanyu Sun, Jiao Xu, Houwen Peng, Dong Wang, Huchuan Lu, and Kede Ma. RELO: Reinforcement learning to localize for visual object tracking. In *ICML*, 2026a.

Zhiming Chen, Xue Wang, Xiaojiang Wu, Yunhua Wu, and Bing Hua. Online decision switching for anti-UAV tracking based on reinforcement learning. *Aerospace Science and Technology*, 2026b. doi: 10.1016/j.ast.2026.113197.

Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. MixFormer: End-to-end tracking with iterative mixed attention. In *CVPR*, 2022.

Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In *CVPR*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *ECCV*, pp. 370–386, 2018.

Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019.

Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *CVPR*, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Bo Huang, Jianan Li, Junjie Chen, Gang Wang, Jian Zhao, and Tingfa Xu. Anti-UAV410: A thermal infrared benchmark and customized scheme for tracking drones in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. GlobalTrack: A simple and strong baseline for long-term tracking. In *AAAI*, 2020.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Yuqing Huang, Xin Li, Zikun Zhou, Yaowei Wang, Zhenyu He, and Ming-Hsuan Yang. RTracker: Recoverable tracking via PN tree structured memory. In *CVPR*, pp. 19038–19047. IEEE, 2024b.

Nan Jiang, Kuiran Wang, Xiaoke Peng, Xuehui Yu, Qiang Wang, Junliang Xing, Guorong Li, Qixiang Ye, Jianbin Jiao, Zhenjun Han, Jian Zhao, and Zhaoxiang Zhang. Anti-UAV: A large-scale benchmark for vision-based UAV tracking. *IEEE Transactions on Multimedia*, 2023.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics, 2023. Version 8.0.0.

Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

Ben Kang, Jie Zhao, Xin Chen, Wanting Geng, Bin Zhang, Lu Zhang, Dong Wang, and Huchuan Lu. UETrack: A unified and efficient framework for single object tracking. In *CVPR*, 2026.

Minji Kim, Seungkwan Lee, Jungseul Ok, Bohyung Han, and Minsu Cho. Towards sequence-level training for visual tracking. In *ECCV*, 2022.

Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI*, 2017.

Xiaohai Li, Bineng Zhong, Qihua Liang, Guorong Li, Zhiyi Mo, and Shuxiang Song. MambaLCT: Boosting tracking via long-term context state space model. In *AAAI*, 2025a.

Xiaohai Li, Bineng Zhong, Qihua Liang, Zhiyi Mo, Jian Nong, and Shuxiang Song. Dynamic updates for language adaptation in visual-language tracking. In *CVPR*, 2025b.

Xin Li, Yuqing Huang, Zhenyu He, Yaowei Wang, Huchuan Lu, and Ming-Hsuan Yang. CiteTracker: Correlating image and text for visual tracking. In *ICCV*, 2023.

Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. Tracking meets LoRA: Faster training, larger model, stronger performance. In *ECCV*, 2024.

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 *ECCV*, 2014.

Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *ICCV*, 2021.

Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *ECCV*, pp. 445–461, 2016.

Matthias Müller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. VastTrack: Vast category visual object tracking. In *NeurIPS*, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *ICLR*, 2025.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

James Supancic III and Deva Ramanan. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In *ICCV*, 2017.

Jovana Videnovic, Alan Lukezic, and Matej Kristan. A distractor-aware memory for visual object tracking with SAM2. In *CVPR*, 2025.

Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: Visual tracking by re-detection. In *CVPR*, 2020.

Hao Wu, Xudong Wang, Jialiang Zhang, Junlong Tong, Xinghao Chen, Junyan Lin, Yunpu Ma, and Xiaoyu Shen. UTPTrack: Towards simple and unified token pruning for visual tracking. In *CVPR*, 2026.

Qiangqiang Wu, Tianyu Yang, Ziquan Liu, Baoyuan Wu, Ying Shan, and Antoni B. Chan. DropMAE: Masked autoencoders with spatial-attention dropout for tracking tasks. In *CVPR*, 2023.

Jinxia Xie, Bineng Zhong, Zhiyi Mo, Shengping Zhang, Liangtao Shi, Shuxiang Song, and Rongrong Ji. Autoregressive queries for adaptive tracking with spatio-temporal transformers. In *CVPR*, 2024.

Chaocan Xue, Bineng Zhong, Qihua Liang, Yaozong Zheng, Ning Li, Yuanliang Xue, and Shuxiang Song. Similarity-guided layer-adaptive vision transformer for UAV tracking. In *CVPR*, 2025.

Chaocan Xue, Qihua Liang, Bineng Zhong, Yanting Zu, Yuanliang Xue, Haiying Xia, and Shuxiang Song. Toward low-cost yet effective temporal learning for UAV tracking. In *CVPR*, 2026.

Bin Yan, Haojie Zhao, Dong Wang, Huchuan Lu, and Xiaoyun Yang. 'Skimming-Perusal' tracking: A framework for real-time and robust long-term tracking. In *ICCV*, 2019.

Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. SAMURAI: Motion-aware memory for training-free visual object tracking with SAM 2. *IEEE Transactions on Image Processing*, 2026. doi: 10.1109/TIP.2026.3651835.

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *ECCV*, 2022.

Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*, 2017.

Chunhui Zhang, Li Liu, Zhipeng Zhang, Yong Wang, Hao Wen, Xi Zhou, Shiming Ge, and Yanfeng Wang. How far are modern trackers from UAV-Anti-UAV? a million-scale benchmark and new baseline. *arXiv preprint arXiv:2512.07385*, 2025.

Jie Zhao, Jingshu Zhang, Dongdong Li, and Dong Wang. Vision-based anti-UAV detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. ODTrack: Online dense temporal token learning for visual tracking. In *AAAI*, 2024.

Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Haotian Wu, Qinqin Nie, Hao Cheng, Chenfeng Liu, et al. VisDrone-VDT2018: The vision meets drone video detection and tracking challenge results. In *ECCV Workshops*, pp. 496–518, 2018.

A Additional Implementation Details

A.1 Training and Decision Features

Training. For both heads, the positive cells are the token cells covered by the ground-truth box, or the cell containing the box center when the box is smaller than one cell. At each positive cell, the binary cross-entropy uses the IoU between the ground-truth box and the predicted box as its soft label, and both losses are normalized by the number of positive cells. The tracker is trained with AdamW at an initial learning rate of 2×10^{-4} and a weight decay of 0.1, using a two-epoch linear warmup followed by cosine annealing. The decision policy is trained with AdamW for 80 epochs at a learning rate of 10^{-3}, with a batch size of 4,096 and a weight decay of 10^{-4}.

Table A1: Decision features.

Feature	Definition
Tracker reliability	
Tracker confidence	c
Confidence average	$\bar{c} \leftarrow 0.9\,\bar{c} + 0.1\,c$
Candidate information	
Proposal score	c_k
Proposal rank	$\log(1 + \mathrm{rank}_k)$
Proposal count	$\log(1 + K)$
Geometry and temporal context	
Center distance	$\|b_k^c - b^{\mathrm{trk},c}\|_2 / s_{\mathrm{win}}$
Scale ratio	$\log(\sqrt{\|b_k\|}/\sqrt{\|b^{\mathrm{trk}}\|})$
Search-region size	$s_{\mathrm{win}}/100$
Switching recency	$\min(\Delta t_{\mathrm{switch}}, 50)/50$

Decision Features and Policy. Table A1 lists the nine decision features of candidate k in three groups. Tracker reliability is described by the current confidence c and its running average $\bar{c}$, which is updated at every frame and starts at 0. Candidate information consists of the proposal score c_k, the rank of the proposal among the K candidates, and the proposal count. The proposal count is constant because K is fixed to 3, so it acts only as a bias term of the policy. Geometry relates the candidate to the current track through the center distance and the scale ratio, where b^c denotes the box center and $|b|$ the box area. The center distance is normalized by the side length s_{win} of the search region, which also enters as a feature. Temporal context is given by the switching recency, the number of frames since the previous switch, which is clipped at 50 and is set to its maximum value before the first switch. The nine features are concatenated into a vector and mapped by two hidden layers of dimension 64 with ReLU to a single predicted advantage. Each proposal is scored independently with the same weights, and the proposal with the highest prediction is compared with τ (Equation (4)).

Full-Frame Input and Motion Residual. The full frame is padded in the same way as the template and search-region crops, which keeps the target at its original aspect ratio in both branches. The frame is scaled by $s = \min(W_f/w, H_f/h)$ and placed at the center of the $W_f \times H_f$ input. The resolution 280×168 approximately matches the 16:9 aspect ratio of most UAV-Anti-UAV frames, and the padded border stays a few pixels wide. A proposal at (x, y) maps back to the frame by $(x - p_x)/s$ and $(y - p_y)/s$, where (p_x, p_y) is the padding offset. For the motion residual, the temporal offset is Δ=5 frames. Phase correlation estimates a single translation between the two grayscale frames at half resolution, or at quarter resolution for frames wider than 1,000 pixels. Pixels without a valid source after alignment are set to zero. The residual is smoothed by a 5×5 Gaussian filter. Training and inference share the same implementation of this residual computation.

A.2 Replay Collection

Replay is applied at every second decision point. The rollout mean IoU $\rho(b, n)$ averages over frames t+1 to t+n in which the target is annotated as present, skips absent frames, and is truncated at the end of the video, and a decision point is discarded when fewer than three such frames remain. All rollouts from a decision point share the same template, backbone, and tracker parameters, while the template remains fixed after initialization. Each cross-fitting fold contains 700 sequences of the UAV-Anti-UAV training split, and in each round, the tracker trained on one fold collects decisions on 450 sequences of the other fold. Table A2 reports the number of labeled decisions and the AUC of the policy fitted on the samples accumulated up to each round. The criteria that activate a decision point are the same in every round, and the rounds differ only in which policy decides whether to switch. The first round uses the hand-crafted rule with the same fixed thresholds as the rule baseline. At a decision point, the features and the advantage labels are computed in the same way regardless of the collecting policy, because the labels come from

Table A2: Replay collection rounds. AUC is measured for the policy fitted on all decisions collected up to that round.

Round	Collection policy	Decisions	Policy AUC
1	Hand-crafted rule	30,453	48.23
2	Learned decision	36,501	48.75
3	Learned decision	31,848	49.07
Total		98,802	

replaying both decisions with the frozen tracker. Switching still changes the subsequent trajectory, so later rounds collect samples from the states that the current policy reaches. On two RTX 5090 GPUs, training each cross-fitting tracker takes approximately 8.4 GPU-hours and the three collection rounds take approximately 5.9 GPU-hours, compared with 8.23 GPU-hours for the final tracker trained on the complete split.

A.3 Inference and Baselines

A decision point is activated when the tracker confidence c falls below $c_{\text{soft}}=0.6$, and periodically every $M=10$ frames regardless of c. Recovery candidates are the K highest-scoring locations of the full-frame score map, without NMS or additional spatial suppression, and their boxes are decoded by the regression MLP of the full-frame head. All base trackers share the same decision threshold $\tau=0.05$, the same number of proposals $K=3$, and the same cooldown of $\Delta_{\text{cool}}=10$ frames.

The external detector of the proposal-source comparison is YOLOv8s (Jocher et al., 2023) with an additional stride-4 detection head, trained for 40 epochs on the UAV-Anti-UAV training split at 960-pixel resolution. Its three input channels are the grayscale current frame, its motion-compensated difference to frame $t-5$, and the aligned frame $t-5$, which provide appearance and motion cues.

B Effect of In-Domain Training

Table A3: Comparison of generic and in-domain training on four anti-UAV benchmarks.

Base	UAV-Anti-UAV		Anti-UAV300		Anti-UAV410		DUT-Anti-UAV	
	Generic	In-domain	Generic	In-domain	Generic	In-domain	Generic	In-domain
LoRAT	37.78	42.22	64.02	64.63	54.05	54.50	60.63	62.68
OSTrack	29.63	37.44	58.29	57.08	42.96	46.94	62.51	61.23
LETrack	28.87	34.89	61.73	62.22	49.99	47.71	61.48	59.24
UTPTrack	27.18	35.40	55.09	56.06	41.09	45.32	63.52	62.05

To assess the gain from in-domain training, we compare four base trackers before and after retraining with the UAV-Anti-UAV training split, without recovery. In-domain training improves AUC on UAV-Anti-UAV by 4.44% to 8.22% for all four base trackers. However, this improvement does not carry over to the other three benchmarks: LoRAT improves by only 0.45% to 2.05%, and OSTrack, LETrack, and UTPTrack all decrease on DUT-Anti-UAV. This gap follows from target size: UAV-Anti-UAV is dominated by small targets, whereas the other three benchmarks contain fewer small targets. Therefore, in-domain training mainly adapts the base tracker to small targets rather than to the larger targets that dominate Anti-UAV300, Anti-UAV410, and DUT-Anti-UAV.

C Results on UAV-Mounted Benchmarks

We further evaluate the recovery decision on UAV-mounted benchmarks, which place the camera on the drone and track ground objects, so rapid camera motion and viewpoint change are more common than long out-of-view intervals. We evaluate the LoRAT-based ReTrack on UAV123 (Mueller et al., 2016), DTB70 (Li & Yeung, 2017), UAVDT (Du et al., 2018), and VisDrone (Zhu et al., 2018) with the same inference configuration as on UAV-Anti-UAV. ReTrack achieves the highest AUC on UAV123, UAVDT, and VisDrone, exceeding never switching by 0.89%, 1.19%, and 0.70% AUC. On DTB70, ReTrack is slightly below never switching, with 66.11% versus 66.52% AUC. On 64% to 79% of the sequences per benchmark, ReTrack never switches, so the two settings give identical trajectories. Therefore, the improvement concentrates on the few sequences in which the base tracker loses the target.

Table A4: Comparison of switching strategies on UAV-mounted benchmarks.

Dataset	Never switch	ReTrack
UAV123	69.07	**69.96**
DTB70	**66.52**	66.11
UAVDT	58.60	**59.79**
VisDrone	62.74	**63.44**

D HAND-CRAFTED RULE AND TRACKER CONFIDENCE

D.1 HAND-CRAFTED RULE

The hand-crafted rule switches to the top proposal when the tracker confidence satisfies $c < \eta_c$, the top proposal score satisfies $c_1 > \eta_p$, and the proposal is displaced from b^{trk} by at least η_d times the search-region side length. Table A5 lists the thresholds of each base tracker. For LoRAT, η_c and η_p are fixed to 0.30 and 0.50. For OSTrack, LETrack, and UTPTrack, we determine the thresholds using one fifth of the training sequences. For each tracker, we run the tracker without switching and record its confidence together with the top proposal score at every frame. We set η_c to the first decile of the recorded tracker confidences and η_p to the median of the recorded top proposal scores over these sequences.

Table A5: Thresholds of the hand-crafted rule.

Tracker	η_c	η_p	η_d
LoRAT	0.30	0.50	0.5
OSTrack	0.44	0.43	0.5
LETrack	0.41	0.24	0.5
UTPTrack	0.44	0.42	0.5

D.2 TRACKER CONFIDENCE

To analyze why the learned decision benefits the base trackers differently, we evaluate how reliably each confidence indicates tracking failures. Without switching, we treat frames with IoU below 0.5 as failures and report the area under the ROC curve (AUROC) of the confidence for detecting them. As shown in Table A6, LoRAT achieves an AUROC at least 0.09 higher than OSTrack, LETrack, and UTPTrack, and ReTrack exceeds the rule by 3.10% AUC on LoRAT and by 1.14% on OSTrack, while it falls below the rule by 0.88% on LETrack and 0.43% on UTPTrack. This difference comes from how the confidence is supervised. LoRAT uses the IoU between the ground-truth and predicted boxes as the soft label of its classification loss, which makes its confidence reflect localization quality. In contrast, OSTrack, LETrack, and UTPTrack supervise a Gaussian heatmap centered on the target, which makes their confidence reflect how close a location is to the target center rather than localization quality. Therefore, the learned decision gains the most on LoRAT, where the confidence better indicates tracking failures, and stays close to the rule on the three heatmap-based trackers.

Table A6: Comparison of tracker confidence on UAV-Anti-UAV.

Tracker	Supervision	AUROC	AUC	
			Rule	ReTrack
LoRAT	IoU label	0.94	45.97	49.07
OSTrack	Heatmap	0.82	41.98	43.12
LETrack	Heatmap	0.85	37.96	37.08
UTPTrack	Heatmap	0.74	41.16	40.73