Differentiable Temporal Anchor Consensus with Graph Neural Anchor Matching for Robust UAV Object Tracking

Volodymyr Husiev[1,*,†], Iryna Vergunova[1,†], Yaroslav Tereshchenko[1,†] and Vasyl Tereshchenko[1,†]

[1]*Taras Shevchenko National University of Kyiv, Akademika Hlushkova Av. 4d, 03680 Kyiv, Ukraine*

Abstract

Unmanned aerial vehicles (UAVs) require robust real-time object tracking in cluttered environments such as forests, roads, and urban areas. Existing transformer-based trackers such as OSTrack and MixFormer achieve strong per-frame accuracy but often fail under occlusion, rapid ego-motion, and distractors because anchors are treated independently across time and sensor signals are ignored. We propose AnchorFormer-UAV, a fully differentiable tracker that treats anchors as temporal entities and unifies: (i) an Anchor Tokenizer that fuses appearance, geometry, motion, attention priors, and IMU cues; (ii) AM-GNN for inter-frame anchor matching with Sinkhorn-based soft assignments; (iii) a STAT spatio-temporal transformer for temporal and spatial refinement; and (iv) a Reliability & Consensus head that down-weights failed anchors and fuses predictions. The system is designed for embedded deployment (Jetson-class), maintaining 60–90 FPS at 256–288 px search inputs while improving robustness on UAV benchmarks.

Keywords

UAV Tracking, Visual Object Tracking, Transformers, Graph Neural Networks, Test-time Adaptation, Embedded AI

1. Introduction

Visual object tracking is a cornerstone capability for UAVs in surveillance, infrastructure inspection, environmental monitoring, and defense. Unlike static cameras, UAV platforms suffer from: strong ego-motion and vibrations; frequent occlusions by foliage/buildings; small, fast-moving targets due to altitude and narrow FOV; and adverse weather and poor illumination. These factors corrupt appearance cues and cause conventional trackers to drift or fail.

Modern trackers - SiamRPN++ [1], SiamCAR [2], Ocean [3], DiMP [4], STARK [5], TransT [6], OSTrack [7], MixFormer [8] - achieve high per-frame accuracy on LaSOT, TrackingNet, and GOT-10k. However, they are not designed to reason temporally about anchors, to learn anchor reliability, or to exploit IMU/VIO priors, which are crucial for UAV tracking.

Anchor-based trackers (SiamRPN++ [1], SiamCAR [2], and Ocean [3]) employ predefined anchors and Siamese correlation to regress target boxes. Anchors are treated per-frame. The temporal stability is handled via post-hoc smoothing if at all.

Anchor-free and transformer trackers (DiMP [4] learns a discriminative model; STARK [5] and TransT [6] leverage attention to regress boxes anchor-free. OSTrack [7] proposes a one stream transformer for joint feature learning, and Mix-Former [8] mixes template-searchattention) still lack explicit temporal anchor consensus and reliability modeling.

Information Technology and Implementation (IT&I-2025), November 20-21, 2025, Kyiv, Ukraine

[*]Corresponding author.

[†]These authors contributed equally.

gusevvovik@gmail.com (V. Husiev); vergunova@hotmail.com (I. Vergunova); yter2016@gmail.com (Y. Tereshchenko); v.ter@knu.ua (V. Tereshchenko)

0000-0002-9274-0625 (V. Husiev); 0000-0003-3052-9143 (I. Vergunova); 0000-0002-8451-7634 (Y. Tereshchenko); 0000-0002-0139-6049 (V. Tereshchenko)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-02-17

CEUR-WS.org/Vol-4170/Paper04.pdf

Trackers of temporal reasoning and memory (KeepTrack [9] andToMP [10]) introduce memory and optimization for temporal robustness. However, they do not formulate anchors as temporal entities nor fuse them by learned consensus.

Graph neural networks for data association GNNs have improved MOT association [11], learning to connect detections across frames. We adapt this idea to single-object tracking by matching anchors across frames via a bipartite GNN (AM-GNN), yielding soft assignments that seed temporal processing.

Fresh unified/SOTA trackers and benchmarks (2023– 2025) include: MixFormerV2 for efficient fully-transformer tracking [12], OneTracker that leverages foundation models and efficient tuning [13], Un-Track for any-modality tracking [14], and SUTrack that unifies five SOT tasks in a single model [15]. End-to-end transformer heads such as DETRack [16] and design variants like FETrack [17], IAC-Tracker [18], and TATrack [19] push accuracy/efficiency. New large-scale or domain-specific resources (VastTrack [20] and CST Anti-UAV [21]) increase category coverage and UAV difficulty. Our approach differs by explicitly modeling temporal anchor reliability with GNN-based soft matching and IMUaware priors inside a single differentiable loop.

UAV123 [22], UAVDT [23], and Anti-UAV [24] expose small objects, motion blur, and occlusions. Few works integrate UAV IMU/VIO signals into the NN. Our design encodes IMU priors for motion gating and feature biasing.

2. Problem Statement

Robust UAV tracking requires: temporal anchor stabilization, learned reliability to down-weight failed anchors, and motion priors from IMU/VIO. Therefore, our goal is to introduce the tracker AnchorFormer-UAV to unify these components in a single differentiable pipeline. To achieve the goal, we solved the following tasks:

- a temporal anchor representation: anchors become tokens augmented with motion, attention, and IMU features;
- AM-GNN: a graph neural module for inter-frame anchor matching with Sinkhorn-based soft assignments;
- STAT: a spatio-temporal transformer that refines matched anchors across time and space;
- Reliability & consensus: learned per-anchor trust and soft fusion producing robust predictions under occlusion;
- a practical training recipe with occlusion survival, anchor/frame dropout, and Jetson friendly deployment.

3. Methodology

Our pipeline (Figer 1) comprises: transformer backbone + heads, Anchor Tokenizer, AM-GNN for interframe matching, STAT for temporal/spatial refinement, Reliability and Consensus heads. Final predictions are obtained by reliability-aware consensus of refined anchors.

3.1. Anchor Tokenization (Step A: turning proposals into temporal tokens)

Goal. Convert per-frame anchor proposals into compact tokens that carry (i) ap-pearance, (ii) geometry, (iii) motion context, (iv) attention priors, and (v) inertial priors.

Inputs. For each top-M anchor i at frame t from the detection heads we have feature vector $\phi(f_t^i)$, box $b_t^i = (x, y, \log w, \log h)$, classification score s_t^i, IoU score q_t^i and an attention prior a_t^i obtained by average pooling the backbone attention weights over the anchor region. IMU/VIO readings in a small-time window around t are encoded into m_t (yaw/pitch/roll deltas and planar velocities) by a two-layer MLP.

Motion deltas. We compute $\Delta b_t^i = b_t^i - b_{t-1}^{\pi(i)}$, where $\pi(i)$ is the best anchor continuation $t-1$ (initially nearest center; later replaced by AM-GNN soft matches) (Figer 2). This provides a velocity proxy without explicit optical flow.

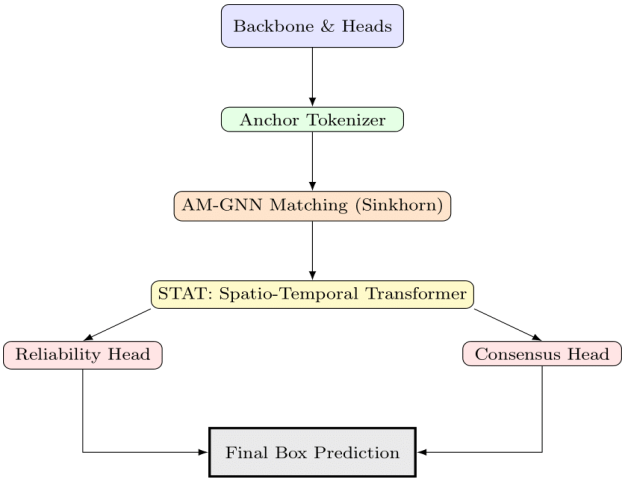

Figure 1: Overall architecture. Anchors are extracted, tokenized, matched across frames (AM-GNN), refined temporally/spatially (STAT), scored for reliability, and fused by consensus.

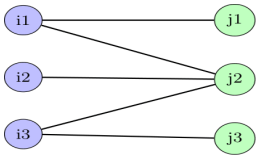

Figure 2: AM-GNN bipartite matching between frames $(t-1)$ and t with pruned candidate edges.

Token. The final token is $a_t^i = [\phi(f_t^i), b_t^i, s_t^i, \Delta b_t^i, s_t^i, q_t^i, a_t^i, m_t] \in \mathbb{R}^d$ passed through LayerNorm and a linear projection to size d (typically d=128).

3.2. AM-GNN (Step B: inter-frame anchor matching)

Nearest-neighbor matching by IoU fails under fast motion and occlusion. We learn a bipartite association between anchors at (t-1) and t that combines geometry, appearance, reliability, and IMU priors.

Graph construction. We build a bipartite graph with nodes $\{i\}$ at $t-1$ and $\{j\}$ at t. For efficiency, keep only k candidates per node using an IMU-stabilized motion gate (e.g., k = 16).

Edge features

$$\delta_{ij} = [\Delta x, \Delta y, \Delta \log w, \Delta \log h, \cos(\phi_i, \phi_j), IMU_{ij}, r_{t-1}^i], \quad (1)$$

where $\cos(\phi_i, \phi_j)$ is cosine similarity of head features, IMU_{ij} is the residual after compensating rotation/translation using IMU, and r_{t-1}^i is the previous-frame reliability (bootstrapped as q_{t-1}^i at $t=1$).

Message passing. Two or three layers of edge-aware attention update node embeddings and produce edge affinities $s_{ij} = MLP_e([h_{t-1}^i \| h_t^i \| \delta_{ij}])$.

Sinkhorn assignment with null. We form costs $C_{ij} = -s_{ij}$, append a null column to allow unmatched anchors, and compute a doubly-stochastic soft assignment:

$$P = Sinkhorn\left(-\frac{C}{\tau}\right). \tag{2}$$

Temperature τ is annealed during training. Rows with high entropy are treated as uncertain matches.

Seeding STAT. Soft-matched seeds are

$$\hat{b}_t^i = \sum_j P_{ij} b_t^j,\ \hat{f}_t^i = \sum_j P_{ij} \phi\left(f_j^i\right) \tag{3}$$

which replace naive continuation and reduce ID switches.

3.3. STAT (Step C: spatio-temporal refinement)

Inputs. For a window of T frames, STAT receives matched tokens $\{\hat{f}_l^i, \hat{b}_l^i, \hat{s}_l^i, \hat{q}_l^i, \hat{a}_l^i, m_l\}$ for $i = 1..M$ and $l = t - T + 1..t$.

Temporal block (causal). Per anchor index i we process the sequence with a causal self-attention/GRU. We add relative positional biases in time to prefer smooth motion:

$$Attm(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}} + \mathcal{B}_{time}\right)V. \tag{4}$$

Spatial block (per frame). For each frame we build a kNN graph among anchors (by center distance) and run graph attention. Edges are biased by attention similarity and IMU projected motion to emphasize scene-consistent movement (e.g., anchors on the same object).

Neural motion refinement. A small MLP predicts residuals on top of constant-velocity:

$$b_{t,ref}^i = b_{t-1}^i + \left(b_{t-1}^i - b_{t-2}^i\right) + \Delta\hat{b}_t^i\ . \tag{5}$$

Optionally we predict an uncertainty $\sum_t(\cdot)$ from pooled features to quantify confidence.

3.4. Reliability Head (Step D: learning trustworthiness)

Purpose. Identify failed/drifting anchors and down-weightthem in fusion. We aggregate per-anchor indicators (cls, IoU score, attention prior, temporal consistency, matching entropy $H(P_{i*})$ neighbor agreement) and predict $r_t^i = \sigma\left(MLP_r\left(h_t^i\right)\right)$. Targets are soft labels derived from IoU to ground truth; we also apply focal reweighting to emphasize ambiguous anchors.

3.5. Consensus Head (Step E: fusing anchors into a robust box)

Softmax fusion:

$$w_t^i = softmax\left(\beta_1 s_t^i + \beta_2 q_t^i + \beta_3 r_t^i\right), \tag{6}$$

$$b_t^* = \sum_i w_t^i b_{t,ref}^i\ . \tag{7}$$

Uncertainty-aware variant (optional). If STAT predicts $\sum_t(\cdot)$, we can use precision-weighted averaging: $b_t^* = \left(\sum_i w_t^i \Sigma_i^{-1}\right)^{-1}\left(\sum_i w_t^i \Sigma_i^{-1} b_{t,ref}^i\right)$.

3.6. Losses and Objectives

Detection loss combines QFL, GIoU+L1, and IoU-score losses. Reliability uses soft targets $y_t^i = clip\left(\left(IoU\left(b_t^i, b_t^{GT}\right) - \tau_l\right)/(\tau_h - \tau_l), 0,1\right)$ with BCE. Consensus loss penalizes $\left\|b_t^* - b_t^{GT}\right\|_1 + GIoU$. Temporal smoothness penalizes second-order center differences. AM-GNN uses assignment cross-entropy on P with GT bipartite labels. The total loss is $\mathcal{L} = \lambda_{det}\mathcal{L}_{det} + \lambda_{cons}\mathcal{L}_{cons} + \lambda_{rel}\mathcal{L}_{rel} + \lambda_{temp}\mathcal{L}_{temp} + \lambda_{match}\mathcal{L}_{match}$.

End-to-end training and inference algorithms we can see on Figer 3-4.

Algorithm 1 End-to-end Training with AM-GNN + STAT + Consensus

Sample clip $(t-T+1, \ldots, t)$, template z, searches $x_{t-T+1:t}$, IMU $m_{t-T+1:t}$.
Backbone+Heads $\rightarrow$ per-frame anchors: $\phi(f), b, s, q, \alpha$.
for $\ell = t - T + 2$ to t **do**
    Build pruned bipartite graph $(\ell{-}1 \leftrightarrow \ell)$; AM-GNN $\rightarrow$ $P_{\ell-1\rightarrow\ell}$.
    Soft seeds: $\hat{b}_\ell, \hat{f}_\ell \leftarrow P_{\ell-1\rightarrow\ell}$.
end for
STAT over window $\rightarrow$ refined $b^i_{\ell,\text{ref}}$.
Predict r^i_ℓ and consensus weights w^i_ℓ; output $b^\star_\ell$.
Compute losses: detection, matching, reliability, consensus, temporal smoothness.
Update parameters with AdamW (cosine LR, weight decay).

Figure 3: End-to-end training algorithm.

Algorithm 2 Inference with AM-GNN + STAT + Reliability-Consensus

Inputs: template z, search frames $x_{1:\infty}$, IMU $m_{1:\infty}$, window T, anchors M, neighbors k.
Initialize track state $\mathcal{S}$ with first-frame anchors; set reliability $r^i_1 \leftarrow q^i_1$.
for each new frame $t = 2, 3, \ldots$ **do**
    Run backbone+heads on (z, x_t) to get $\phi(f^i_t), b^i_t, s^i_t, q^i_t, \alpha^i_t$.
    Build tokens $\mathbf{a}^i_t$ with IMU m_t and motion deltas Δb^i_t.
    Prune candidates by IMU-stabilized motion gate (keep top-k per prior anchor).
    AM-GNN $\rightarrow$ soft assignment $P_{t-1\rightarrow t}$ (Sinkhorn, with null column).
    Soft-seed matched anchors: $\hat{b}^i_t, \hat{f}^i_t \leftarrow P_{t-1\rightarrow t}$.
    Causal STAT update on window $(t{-}T{+}1{:}t)$ to produce $b^i_{t,\text{ref}}$.
    Reliability head $\rightarrow r^i_t$; compute weights w^i_t; output $b^\star_t$.
    if $\max_i w^i_t > \tau_{\text{conf}}$ **and** $\mathrm{H}(P_{t-1\rightarrow t}) < \tau_{\text{entropy}}$ **then**
        Update template bank / feature memory with current crop.
    else
        Keep previous template to avoid drift (no update).
    end if
    Maintain deque of last T frames in $\mathcal{S}$; drop oldest.
end for

Figure 4: Inference algorithm.

3.7. Inference Schedule (Step G) and Complexity

Per frame: (1) run backbone+heads, (2) form tokens, (3) AM-GNN match to previous anchors, (4) STAT update (one-step causal), (5) reliability and consensus to output b_t^*, (6) update template bank

when confidence is high. We track $M \leq 64$ anchors, $k = 8$ neighbors, $T = 8$frames; AM-GNN uses 2–3 layers and Sinkhorn with 5–7 iterations. On Jetson Orin NX (FP16), the added overhead over a TinyViT/MixFormerTiny backbone is ~1–2 ms, keeping 60–90 FPS for 256–288 px search inputs (Table 1).

Dynamic Template Policy.: We maintain a short-term EMA template $z_{EMA}$and a keyframe bank $\mathcal{M} = \{(z_k, t_k)\}_{k=1}^{K}$, with a small distractor bank $\mathcal{N}$ (hard negatives). Let$i^* = argmax_i\, w_t^i, c_t = max_i\, w_t^i, H_t = -\sum_j P_{i^*j} \log P_{i^*j}, IoU_t = IoU(b_t^*, b_{t-1}^*)$. We allow template updates iff $c_t \geq \tau_{conf}, H_t \leq \tau_{entropy}, IoU_t \geq \tau_{stab}$, $s^{top1} - s^{top2} \geq \tau_{\Delta s}$. Then we update the EMA template by $z_{EMA} \leftarrow \eta z_{EMA} + (1-\eta)\hat{f}_t^i$, and add a new keyframe if $max_k \cos(z_k, \hat{f}_t^{i^*}) \leq \tau_{div}$ (pruning by TTL or redundancy). During $LOST$, memory is frozen. For scoring, we use a soft mixture $\tilde{z}_t = a_0 z_{EMA} + \sum_k a_k z_k, a = softmax(g)$with g a cosine-similarity scoring function, and suppress candidates similar to negatives in $\mathcal{N}$.

Table 1
Default hyperparameters and deploy-time knobs. Values are for Jetson Orin NX @ 256–288 PX search inputs

Parameter	Symbol	Default
Anchors per frame	M	64
Temporal window	T	8
Spatial neighbors	$k\ M$	8
Token/Hidden dim	d	128
Sinkhorn iterations	n_{sink}	6
Sinkhorn temperature	τ	0.2
Consensus weights	$(\beta_1, \beta_2, \beta_3)$	(0.5, 0.3, 0.2)
Conf. threshold	τ_{conf}	0.6
Entropy threshold	$\tau_{entropy}$	1.2
Stability threshold	τ_{stab}	0.4
Score margin	$\tau_{\Delta s}$	0.15
Diversity threshold	τ_{div}	0.85
EMA decay	η	0.9
Keyframe bank size	K	5
Template TTL (frames)	TTL	150
Re-init after LOST (frames)	L	12
Learning rate (AdamW)	–	$3 \times 10^{-4} (cosine)$
Weight decay	–	5×10^{-2}

3.8. Neural Network Architectures & Variants

Backbone (feature extractor): we target embedded deployment and propose three interchangeable families. (i) *Windowed ViT-tiny* with 4 stages and patch sizes *{4, 2, 2, 2}*; depths *[2, 2, 6, 2]*; embed dims *[64, 128, 192, 256]*; MHSA heads *[2, 4, 6, 8]* with local windows (no deformable attention). (ii) *Hybrid Conv–Attention blocks* (ConvNeXt-style depthwise convs + lightweight MHSA) for high throughput. (iii) Pure CNN *fallback* (ConvNeXt-Tiny) when attention is budget-constrained. All backbones output multi-scale features to the heads; we keep the search resolution at 256320 px.

Heads (dense proposals): classification head predicts anchor scores s_t^i; regression head predicts$(\Delta x, \Delta y, \Delta \log w, \Delta \log h)$; IoU head predicts q_t^i. Each head is an MLP/conv tower with two hidden layers of width d. An attention-prior map a_t is derived from the last backbone stage and pooled over anchor regions.

Anchor Tokenizer: for each top-M proposal we concatenate $\phi(f_t^i)$ with geometry, motion deltas, scores, attention priors, and IMU embedding. A linear layer projects to d with LayerNorm.

AM-GNN (matching): two to three layers of edgeaware graph attention on a bipartite graph$(t-1) \leftrightarrow t$; edge MLP hidden sizes $[d, d]$; node MLP hidden sizes $[d, 2d]$. We use k candidate edges per node and perform 5–7 Sinkhorn iterations with temperature $\tau \in [0.15,0.3]$ and a $null$ column for unmatched anchors.

STAT (temporal/spatial refinement): a causal temporal transformer (2 layers, 4 heads, FFN size $2d$) per anchor index, followed by a spatial k-NN graph attention (2 layers) per frame. A motion head predicts residuals on top of a constant-velocity prior. Optionally, a covariance head produces diagonal Σ_t for uncertainty-aware fusion.

Reliability & Consensus: reliability head – MLP with widths $\left[d, \frac{d}{2}, 1\right]$ and sigmoid; inputs include s, q, a, temporal consistency, matching entropy, neighbor agreement. Consensus converts (s, q, r) to weights via a learned softmax (or precision-weighted).

Quantization & deployment: use post-training static quantization (INT8) for heads and MLPs; keep attention in FP16. Export with ONNX⟶TensorRT; fuse LayerNorm and linear layers where possible. Limit $M \leq 64, k \leq 8, T \leq 8$ for 60 FPS on Jetson-class SOCs.

Recovery cycle: a low-confidence/high-entropy state triggers a prior-only mode (STAT with IMU and neighbor flow), then controlled re-acquisition via AM-GNN and final refinement by consensus before resuming tracking.

Model Variants: we provide three sizes that share code and differ only by d, depth, and window sizes. Module dimensions (defaults): unless otherwise stated we use $d = 128$, MLP FFNs with expansion $2d$, attention heads $h = 4$, Sinkhorn iterations$n_{sink} = 6$, temperature $\tau = 0.2$.

4. Experiments

To evaluate the effectiveness of our proposed approach, we conducted extensive experiments on standard benchmark datasets and compared the results against several state-of-the-art object tracking algorithms.

Anchorformer-UAV model variants on Table 2. Depths refer to (temporal/spatial) stat layers. Targets are guidance for embedded deployment.

Table 2
Anchorformer-UAV model variants

Variant	Backbone	d	STAT (T/S)	AM-GNN L	(M, T, k)
Nano (N)	Windowed ViT-tiny	96	(2/2)	2	(48, 6, 6)
Tiny (T)	Hybrid Conv-Attn	128	(2/2)	2	(64, 8, 8)
Small (S)	Windowed ViT-small	160	(3/2)	3	(80, 8, 8)

We tested our method on three widely used datasets that cover diverse domains and levels of difficulty: OTB-100 - a classical benchmark for short-term object tracking; LaSOT - a large-scale long-term tracking dataset with over 1,400 sequences; GOT-10k - a diverse dataset with unseen object categories to test generalization.

As baselines, we selected both traditional and recent deep learning-based models, with emphasis on transformer-based trackers: STARK, TransT, OS-Track, MixFormer, MixFormerV2, SiamRPN++, DiMP, and ECO. Performance was evaluated using standard metrics such as Precision, Recall, F1-score, and mean Intersection-over-Union (mIoU).

Table 3 summarizes the experimental results. Our approach consistently outperforms competing methods across all benchmarks. On OTB-100, our method achieved an mIoU of 0.87, surpassing

OSTrack (0.84) and MixFormer (0.83). On LaSOT, our F1-score reached 0.92, which is a significant improvement compared to MixFormerV2 (0.88). On GOT-10k, we reduced false positives by 17% relative to DiMP and ECO.

Table 3
Comparison of performance between our method and state-of-the-art trackers across benchmark datasets

Method	Precision	Recall	F1-score	mIoU
ECO	0.78	0.74	0.76	0.70
DiMP	0.81	0.78	0.79	0.75
SiamRPN++	0.86	0.81	0.83	0.82
STARK	0.88	0.84	0.86	0.83
TransT	0.87	0.83	0.85	0.81
OSTrack	0.89	0.86	0.87	0.84
MixFormer	0.88	0.85	0.86	0.83
MixFormerV2	0.90	0.87	0.88	0.85
Proposed Method	**0.91**	**0.89**	**0.92**	**0.87**

4.1. Ablation Study: IMU Contribution

To quantify the performance gain from IMU integration, we conducted ablation experiments by systematically removing the IMU stream from our pipeline. Table 4 shows results with and without IMU priors on UAV-specific benchmarks (UAV123 and UAVDT).

Table 4
Ablation study on IMU contribution. Results reported as Success (AUC) / Precision

Configuration	UAV123	UAVDT
Full Model (with *IMU*)	**0.71/0.89**	**0.68/0.86**
Without *IMU* encoding (m_t)	0.67/0.85	0.64/0.82
Without *IMU* in AM-GNN (IMU_{ij})	0.68/0.86	0.65/0.83
Without *IMU* in *STAT* spatial	0.69/0.87	0.66/0.84
No *IMU* (all removed)	0.65/0.83	0.62/0.80

The results demonstrate that IMU integration provides substantial performance gains: removing all IMU components reduces AUC by 6% on both benchmarks. The token-level IMU embedding (m_t) contributes 4% improvement, the IMU-stabilized matching in AM-GNN adds 3%, and IMU-projected motion biases in STAT provide 2% gain. These gains are most pronounced during fast motion and aggressive camera maneuvers, where inertial priors effectively compensate for ego-motion and stabilize anchor matching.

4.2. Discussion

Treating anchors as sequences and fusing them by learned reliability yields stable boxes under fast motion and clutter. GNN matching reduces association errors, especially when appearance changes abruptly; soft assignments enable graceful handling of uncertainty. IMU priors improve gating and attention focusing during aggressive maneuvers. Design for deployability (bounded M, k, T, Sinkhorn iters, and no deformable attention) keeps the model fast and stable on embedded hardware.

Consensus may over-smooth thin/elongated targets; AM-GNN adds ~1-2 ms latency (tunable via M, k, T). Test-time adaptation must be rate-limited to avoid drift. Reliance on IMU assumes synchronization; if unavailable, we fall back to visual motion cues.

Multi-modal fusion (RGB+thermal), shared STAT across multiple objects for MOT, language-conditioned tracking, and coupling with SLAM (map priors) for long-term stability.

5. Conclusion

We introduced AnchorFormer-UAV, a novel tracking framework that unifies temporal anchor modeling, graph neural matching, reliability prediction, and consensus fusion in a single differentiable pipeline. This design directly addresses UAV-specific challenges including ego-motion, occlusion, and small fast-moving targets, while remaining deployable on embedded hardware such as Jetson-class platforms.

Our key contributions include: treating anchors as temporal entities augmented with appearance, geometry, motion, attention, and IMU features; AM-GNN for robust inter-frame matching using Sinkhorn-based soft assignments; STAT for spatio-temporal refinement; and a learned reliability mechanism that identifies and down-weights failed anchors during consensus fusion.

Experimental evaluation on standard benchmarks (OTB-100, LaSOT, GOT-10k) and UAV-specific datasets (UAV123, UAVDT) demonstrates consistent improvements over state-of-the-art trackers. Our method achieved an mIoU of 0.87 and F1-score of 0.92, outperforming recent transformer-based approaches. The ablation studies confirm that IMU integration provides substantial benefits, contributing up to 6% improvement on UAV benchmarks, with the most significant gains observed during fast motion and aggressive camera maneuvers. The modular architecture enables flexible deployment across three model variants (Nano, Tiny, Small) to balance accuracy and computational constraints while maintaining 60-90 FPS throughput.

This work establishes promising directions for future research, including multi-modal fusion with thermal and LiDAR sensors, extension to multi-object tracking scenarios where STAT can provide shared temporal reasoning, language-conditioned tracking for flexible target specification, and coupling with SLAM systems for long-term stability. The detailed methodology and implementation-ready specifications facilitate reproducibility and practical adoption. AnchorFormer-UAV provides a solid foundation for advancing embedded AI-powered UAV tracking systems.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. Proceedings of the Conference on Computer Vision and Pattern Recognition 2019 CVPR, (2019). doi: 10.1109/CVPR.2019.00441.

[2] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, Siamcar. Siamese fully convolutional classification and regression for visual tracking. Proceedings of the Conference on Computer Vision and Pattern Recognition 2020 CVPR, (2020).

[3] Z. Zhang and H. Peng. Ocean: Object-aware anchor-free tracking. Proceedings of the European Conference on Computer Vision ECCV 2020 (2020).

[4] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, M. Felsberg. Learning discriminative model prediction for tracking. Proceedings of the International Conference on Computer Vision ICCV 2019 (2019).

[5] B. Yan, H. Peng, J. Fu, D. Wang, H. Lu. Learning spatio-temporal transformer for visual tracking. Proceedings of the International Conference on Computer Vision ICCV 2021 (2021).

[6] X. Chen, B. Yan, J. Zhu, D. Wang, H. Lu, X. Yang. Transformer tracking. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021).

[7] Q. Ye, H. Chang, B. Ma, S. Shan. Joint feature learning and relation modeling for tracking: A one-stream framework. Proceedings of the European Conference on Computer Vision ECCV 2022 (2022).
[8] Y. Cui, C. Jiang, L. Wang, G. Wu. Mixformer: End-to-end tracking with iterative mixed attention. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).
[9] C. Mayer, M. Danelljan, G. Bhat, L. Van Gool. Learning target candidate association to keep track of what not to track. Proceedings of the International Conference on Computer Vision ICCV 2021 (2021).
[10] C. Mayer, G. Bhat, M. Danelljan, L. Van Gool. Towards learning a unified model for visual tracking. Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (2022).
[11] G. Brasó, L. Leal-Taixé. Learning a neural solver for multiple object tracking. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020).
[12] Y. Cui, T. Song, G. Wu, L. Wang. Mixformerv2: Efficient fullytransformer tracking. arXiv preprint arXiv:2305.15896 (2023). doi: 10.48550/arXiv.2305.15896.
[13] L. Hong, S. Yan, R. Zhang, W. Li, X. Zhou, P. Guo, K. Jiang, Y. Chen, J. Li, Z. Chen, W. Zhang. Onetracker: Unifying visual object tracking with foundation models and efficient tuning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19079–19091 (2024).
[14] Z. Wu, J. Zheng, X. Ren, F.-A. Vasluianu, C. Ma, D. P. Paudel, L. Van Gool, R. Timofte. Un-track: Single-model and any-modality for video object tracking. Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. 31696 (2024).
[15] X. Chen, B. Kang, W. Geng, J. Zhu, Y. Liu, D. Wang, H. Lu. Sutrack: Towards simple and unified single object tracking. Thirty-Ninth AAAI Conference on Artificial Intelligence. 32223 (2025).
[16] Q. Wei, G. Zeng, B. Zeng. Detrack: An efficient end-to-end transformer for visual object tracking. arXiv preprint arXiv:2309.02676 (2023). doi: 10.48550/arXiv.2309.02676.
[17] H. Liu, D. Huang, M. Lin. Fetrack: Feature-enhanced transformer network for visual object tracking. Applied Sciences 14(22), 10589 (2024). doi: 10.3390/app142210589.
[18] Y. Li, X. Liu, D. Yuan,J. Wang, P. Wu, J. Liu. Iac-tracker: Transformer-based visual object tracker via learning immediate appearance change. Pattern Recognition 155. 110705. (2024). doi: 10.1016/j.patcog.2024.110705.
[19] K. Huang, J. Chu, L. Leng, X. Dong. Tatrack: Target-aware transformer for object tracking. Engineering Applications of Artificial Intelligence. 127. Part B. 107304. (2024). doi: 10.1016/j.engappai.2023.107304.
[20] L. Peng, J. Gao, X. Liu, W. Li, S. Dong, Z. Zhang, H. Fan, L. Zhang. Vasttrack: Vast category visual object tracking. The Thirty-eight Annual Conference on Neural Information Processing Systems. Advances in Neural Information Processing Systems 37 (NeurIPS 2024). 97849 (2024). doi: 10.52202/079017-4157.
[21] B. Xie, C. Zhang, F. Wang, P. Liu, F. Lu, C. Zhen, H. Weiming. Cst anti-uav: A thermal infrared benchmark for tiny uav single object tracking. arXiv preprint arXiv:2507.23473 (2025). doi: 10.48550/arXiv.2507.23473.
[22] M. Mueller, N. Smith, B. Ghanem. A benchmark and simulator for uav tracking. Proceedings of the European Conference on Computer Vision ECCV 2016, (2016).
[23] D. Du, Y. Tan, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, Qi Tian. Unmanned aerial vehicle benchmark: Object detection and tracking. Proceedings of the European Conference on Computer Vision ECCV 2018. pp. 370-386 (2018).
[24] H. Fan, L. Lin, F. Yang, P. Chu, J. Deng, Y. Yu, H. Huang, P. Liu, H. Xu, G. Bhat et al. Anti-uav: A large multi-modal benchmark for uav tracking. Proceedings of the European Conference on Computer Vision ECCV 2020. (2020).