sensors

MDPI

Article

AeroGRT: Geometry-Aware Global–Local Routing Transformer with Query-Reliability Graphs for UAV Object Recognition in Complex Scenes

Yihua Zang [1,*], Xingmin Ma [1], Zhi Liu [2] and Mengxing Ma [3]

[1] System Department 2, 15th Research Institute of China Electronics Technology Group Corporation, Beijing 100083, China; maxingmin1985@163.com
[2] School of Information, Beijing Institute of Technology, Beijing 100081, China; 7520170087@bit.edu.cn
[3] School of Electrical and Information Engineering, Yunnan Minzu University, Kunming 650031, China; mamengxing1980@163.com
* Correspondence: zangyihua2026@163.com

Highlights

What are the main findings?

- By integrating global–local dual encoding, cross-axis boundary attention, and query-reliability graph attention, AeroGRT achieved 75.8% average overlap (AO), 83.9% SR0.5, and 73.1% SR0.75 on GOT-10k, improving AO by 12.4 percentage points over the ablation baseline.
- AeroGRT attained 87.4% AUC, 91.3% normalized precision, and 87.6% precision on TrackingNet, while improving mAP50 by 2.8 and 1.1 percentage points on VisDrone and TinyPerson, respectively, compared with the corresponding YOLOv8-based baselines.

What are the implications of the main findings?

- The synergistic integration of boundary-sensitive feature fusion and feature-consistency-guided attention provides a geometry-aware mechanism for preserving fine-grained target representations and mitigating unreliable foreground–background associations under occlusion, scale variation, and severe visual clutter.
- The proposed scale–altitude routing strategy adaptively redistributes detection confidence across feature-pyramid levels without altering bounding-box geometry, offering a unified architectural foundation for multiscale UAV object perception.

Abstract

UAV visual perception is increasingly important for intelligent transportation monitoring, infrastructure inspection, emergency response, and low-altitude surveillance. However, reliable target localization remains challenging in complex scenes where targets are small, densely distributed, weakly textured, partially occluded, or visually similar to the background. Motivated by attention-based feature fusion and the routing attention design of AnisoNet, this paper proposes AeroGRT, a framework integrating global–local feature encoding and query-reliability graph attention for UAV visual perception in complex scenarios. The proposed framework combines a CNN-based local encoder, a Transformer-based global encoder, an adaptive global optimization module, a foreground–background fusion attention module, a cross-axis boundary attention mechanism, a query-reliability graph attention module, and a scale–altitude routing detection head. The architecture is designed to integrate local spatial detail with long-range semantic context, preserve target boundary information, and reduce unreliable query–key associations between foreground targets and background clutter. AeroGRT was evaluated on GOT-10k and TrackingNet for object

Check for updates

Academic Editor: Ben Hamza

Received: 21 September 2026
Revised: 21 October 2026
Accepted: 22 October 2026
Published: 24 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

tracking and on VisDrone and TinyPerson for small-object detection, using task-specific protocols and metrics. These evaluations assess performance under occlusion, deformation, low contrast, background clutter, and long-range observation. This study investigates the joint modeling of local detail, global context, boundary cues, and query reliability for accurate UAV target localization in challenging environments.

Keywords: UAV image recognition; object detection; transformer; cross-attention; query-reliability graph

1. Introduction

UAV platforms provide flexible low-altitude sensing, wide spatial coverage, and rapid deployment for traffic surveillance, road infrastructure inspection, urban security, agricultural monitoring, and emergency rescue. Public drone benchmarks such as VisDrone, UAVDT, and UAV123 have made it possible to evaluate detection and tracking algorithms under realistic UAV viewpoints, dense traffic scenes, motion blur, scale variation, and severe background clutter [1–3]. Compared with fixed surveillance cameras or ground vehicles, UAV imagery introduces stronger scale instability, larger perspective changes, and more frequent small-target conditions. These characteristics make UAV image recognition a demanding research problem that connects object detection, single-object tracking, multi-object tracking, semantic perception, and edge computing deployment.

The development of deep object detectors has substantially improved image recognition accuracy. Region-based approaches introduced candidate-region reasoning and high-quality localization [4,5], while one-stage detectors represented by the YOLO family advanced real-time recognition for practical deployment [6–9]. More recently, DETR-style detectors and real-time detection Transformers have improved the modeling of long-range dependencies and set-based prediction [10]. Nevertheless, UAV scenes remain challenging because target size, target density, viewpoint, and background texture change simultaneously. A detector that performs well in ground-level images may fail when the same target occupies only a few pixels, is partially occluded by vegetation or vehicles, or is embedded in shadows and textured roads.

Transformer models offer powerful global modeling ability because multi-head self-attention can directly establish dependencies among distant visual tokens [11–17]. In visual tracking, TransT [18] replaces correlation-based template–search feature fusion with an attention-based network comprising an ego-context augment module based on self-attention and a cross-feature augment module based on cross-attention. These developments motivate the investigation of attention-driven feature interaction for UAV visual perception. In this study, we focus on preserving local image details, representing object boundaries, and suppressing unreliable foreground–background associations. These challenges motivate the combination of local feature extraction, global contextual modeling, and query–key consistency modeling in AeroGRT.

Crack-monitoring studies investigate low-contrast and geometrically irregular road defects in ground-based and UAV imagery [19–26]. AnisoNet [27] provides a related example of attention and convolutional designs for anisotropic crack detection. Such research motivates considering target geometry when developing feature representations for UAV images, in which small or elongated objects may have weak contours and limited visual detail. This study therefore investigates the integration of global–local feature encoding, boundary-sensitive attention, query-reliability modeling, and geometry-aware

routing for UAV object localization. Figure 1 summarizes the challenging UAV scenarios considered in this study and their relationship to the AeroGRT framework.

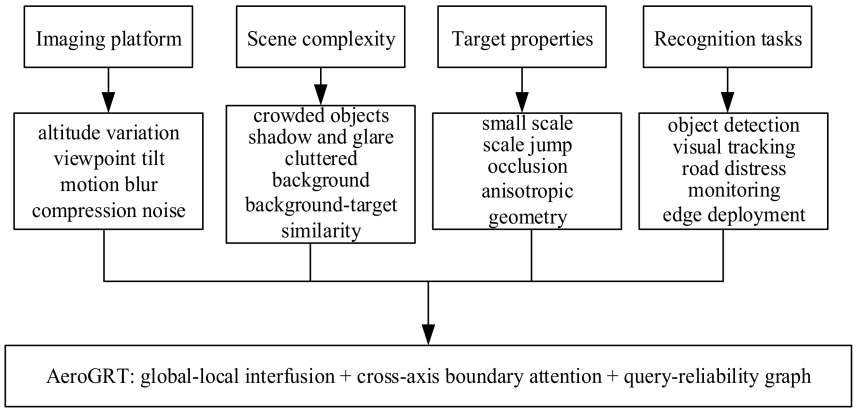


Figure 1. Complex UAV image recognition scenario taxonomy and the corresponding AeroGRT algorithm.

The main contributions are summarized as follows: (1) We propose AeroGRT, an interfusion feature extraction and query-reliability graph Transformer for UAV image recognition in complex scenarios. (2) We design a dual-encoder global–local feature extraction network that preserves local detail while maintaining global semantic context. (3) We formulate a cross-axis boundary attention mechanism to fuse horizontal and vertical local information and enhance boundary-sensitive features. (4) We construct a query-reliability graph attention module that uses feature-space consistency to weight query–key associations.

The remainder of this paper is organized as follows: Section 2 reviews Transformer models, attention mechanisms, and UAV image recognition. Section 3 presents the proposed AeroGRT framework and mathematical formulation. Section 4 reports datasets, implementation settings, ablation studies, comparative experiments, plots, and analysis. Section 5 concludes the paper and discusses future research directions.

2. Related Work

2.1. Transformer Model

The Transformer architecture introduced multi-head attention as a general mechanism for modeling dependencies among sequence elements [11]. Vision Transformer and DeiT demonstrated that image patches could be treated as visual tokens, enabling global visual recognition with large-scale or data-efficient training [12,13]. DETR formulated object detection as set prediction and replaced hand-designed anchors and NMS-heavy pipelines with Transformer decoding [14]. Hierarchical architectures such as PVT, Swin Transformer, SegFormer, and BiFormer improved dense prediction by combining multi-scale representation, shifted windows, efficient routing, and pyramid design [15–17,28].

Despite these advantages, UAV recognition demands additional design constraints. Direct global self-attention may dilute local contour cues and introduce computational overhead. Dynamic snake convolution [29], large-kernel convolution [30], structural reparameterization [31], and convolutional attention [32] have been introduced to recover geometry, efficiency, and local inductive bias. Therefore, an effective UAV recognition architecture should combine CNN-based local perception with Transformer-based global context modeling.

2.2. Attention Mechanism

Attention mechanisms are widely used to model feature interactions in visual tracking. Representative approaches include spatio-temporal Transformer tracking [33], sequence-to-sequence tracking [34], iterative mixed attention [35], shifted-window Transformer tracking [36], and attention-in-attention tracking [37]. UAV-oriented Siamese Transformer pyramids [38] and compact Transformer tracking with correlative masked modeling [39] further extend the range of architectures relevant to this study. A review of Siamese UAV tracking provides broader context for target appearance variation, scale changes, and efficient deployment [40]. These studies motivate attention-based feature interaction for UAV localization and provide relevant tracking architectures for comparison.

Attention mechanisms can benefit from feature complementarity and consistent query–key associations. Feature fusion combines global and local features, cross-axis attention aggregates horizontal and vertical information, and the query-reliability mask weights query–key associations using feature-space consistency. In AeroGRT, these operations are used for global–local scene encoding and multi-level feature fusion for UAV recognition.

2.3. UAV Image Monitoring and Recognition

Research relevant to UAV monitoring spans road-distress analysis, traffic-object recognition, and small-object detection. Civil-infrastructure studies investigate crack detection, pavement-distress classification, and benchmark datasets for road-damage analysis [19–26]. Recent UAV detection research includes cooperative mixed-task training [41], lightweight task-aligned detection [42], self-attention-guided global feature fusion [43], and multi-scale Transformer attention for small-object detection [44]. Zunair et al. introduced RSUD20K, a road-scene object detection dataset containing over 20,000 high-resolution images from Bangladesh, with approximately 130,000 bounding-box annotations across 13 object classes [45]. Its crowded traffic scenes, diverse viewpoints, and varying weather conditions are relevant to the background clutter and appearance variability considered in this study. RSUD20K uses a ground-level driving perspective, whereas AeroGRT targets UAV observations involving altitude-related scale and viewpoint changes. It therefore provides complementary context for road-scene perception. These studies motivate the preservation of fine-scale visual evidence together with contextual feature modeling in complex scenes. Table 1 summarizes representative algorithm families and their conceptual relevance to AeroGRT.

Table 1. Representative algorithm families and their relevance to AeroGRT.

Family	Representative Idea	Limitation in UAV Scenes	AeroGRT Response
One-stage detection	Direct dense prediction with real-time inference	Small objects and dense scenes may be underrepresented	Scale–altitude routing and P2–P5 prediction
DETR-style detection	Global set prediction and long-range interaction	Slow convergence and weak local detail in tiny targets	Dual encoder plus local cross-axis attention
Transformer tracking	Template–search interaction and spatio-temporal attention	False matching under background similarity	Query-reliability graph attention
Road distress monitoring	Anisotropic and low-contrast target perception	Strong geometry but task-specific design	Geometry-adaptive feature routing generalized to UAV objects
Edge UAV recognition	On-board inference and low latency	Accuracy–efficiency trade-off under resource limits	Pyramid-level confidence routing with optional altitude metadata

3. UAV Image Recognition Algorithm Based on Fusion Feature Extraction and Query-Reliability Map Transformation

3.1. Interfusion Feature Extraction Network Model

The overall architecture of AeroGRT is shown in Figure 2. The model receives UAV image frames and optional altitude metadata. The input is first normalized and resized; then a CNN-based local branch extracts fine-grained feature maps, while a Transformer-based global branch models long-range semantic dependencies. The global optimization module forms stage-weighted global representations, and the fusion attention module aligns them with local features at four fusion stages to construct complementary foreground- and background-weighted representations. The resulting features are used by the cross-axis attention module, query-reliability graph module, and scale–altitude routing detection head.

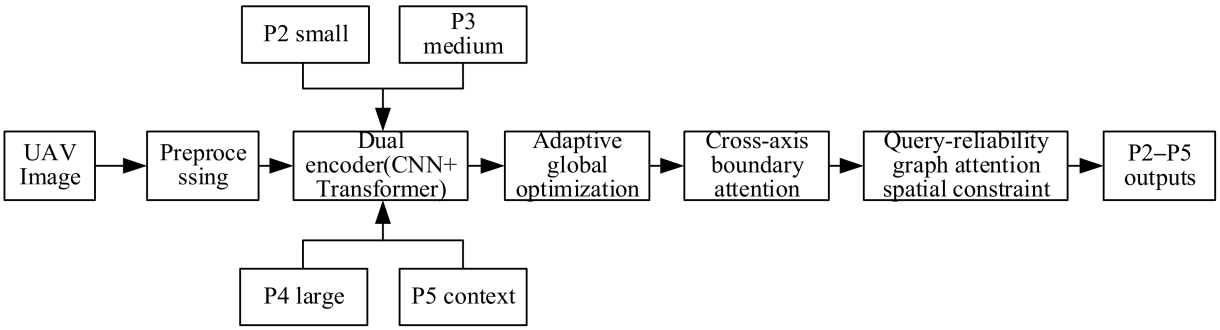


Figure 2. Overall architecture of AeroGRT, including preprocessing, dual-branch feature extraction, global optimization, cross-axis attention, query-reliability graph attention, and scale-altitude routing head.

(1) Complementary Feature Extraction Module

The complementary feature extraction module is shown in Figure 3. Let I_t denote the input UAV image at time t. The global branch partitions I_t into non-overlapping patches and embeds them into a token sequence U_0. After L Transformer layers, the final token sequence U_L represents scene-level relationships. In parallel, the local branch extracts a hierarchy of convolutional feature maps F_1, F_2, F_3, and F_4, which retain local edges, textures, and small-object details.

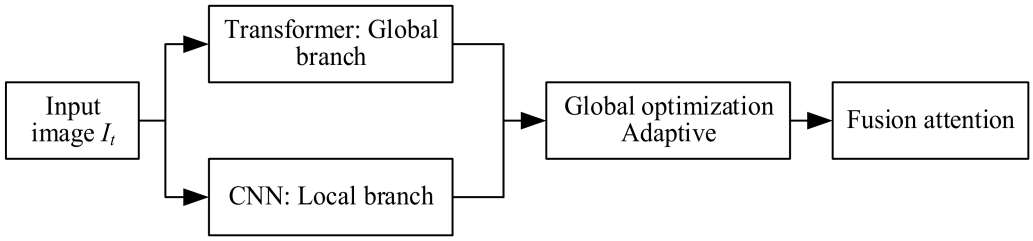


Figure 3. Detailed structure of the interfusion feature extraction network model.

Equation (1) defines patch embedding with positional encoding, Equation (2) describes the Transformer updates, Equation (3) constructs the local feature hierarchy, and Equation (4) collects the outputs of the two branches. The final global sequence $U_L \in \mathbb{R}^{N_{\text{tok}} \times d}$ contains only spatial patch tokens for feature fusion, with no classification token included in this sequence. Here, $N_{\text{tok}} = h_u w_u$, where h_u and w_u denote the height and width of the token grid, respectively, and d denotes the token dimension. The local feature at stage j is $F_j \in \mathbb{R}^{c_j \times h_j \times w_j}$, where c_j, h_j, and w_j denote its channel count, height, and width. The positional encoding E_{pos} has the same shape as the patch embedding. Throughout this

section, Cat denotes concatenation along the channel dimension, and $\odot$ denotes element-wise multiplication with broadcasting where specified.

$$\mathbf{U}_0 = \mathrm{PatchEmbed}(\mathbf{I}_t) + \mathbf{E}_{\mathrm{pos}} \tag{1}$$

$$\mathbf{U}_\ell = \mathcal{T}_\ell(\mathbf{U}_{\ell-1}), \quad \ell = 1, \ldots, L \tag{2}$$

$$\mathbf{F}_j = \mathcal{C}_j(\mathbf{F}_{j-1}), \quad j = 1, \ldots, 4, \qquad \mathbf{F}_0 = \mathbf{I}_t \tag{3}$$

$$\Omega = \{\mathbf{U}_L, \mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4\} \tag{4}$$

(2) Global Optimization Module

The global optimization module provides stage-weighted global representations for the four local–global fusion stages. As shown in Figure 4, AxAttn denotes shape-preserving axial self-attention applied first along the horizontal axis and then along the vertical axis of F_4. Both operations preserve the channel count c_4 and the spatial size $h_4 \times w_4$. The spatial tokens U_L are reshaped into a feature map of size $d \times h_u \times w_u$, and adaptive max pooling produces an aligned feature $\widetilde{U} \in \mathbb{R}^{d \times h_4 \times w_4}$. Equation (5) concatenates AxAttn(F_4) and $\widetilde{U}$ along the channel dimension, giving $Z_{\mathrm{gl}} \in \mathbb{R}^{c_{\mathrm{gl}} \times h_4 \times w_4}$, where $c_{\mathrm{gl}} = c_4 + d$. In Equation (6), each Gate_j is an independent 1×1 convolution mapping c_{gl} channels to a single-channel score map. Global average pooling over the spatial dimensions yields a scalar score e_j, and a softmax over the four stages produces weights satisfying $\sum_{j=1}^{4} \alpha_j = 1$. Equation (7) scales the shared global representation for each fusion stage. The subsequent stage-dependent alignment operator $\mathcal{R}_j$ maps Z_j to the shape of the corresponding local feature.

$$\boldsymbol{Z}_{\mathrm{gl}} = \mathrm{Cat}\left[\mathrm{AxAttn}_{\mathrm{ver}}(\mathrm{AxAttn}_{\mathrm{hor}}(F_4)), \mathrm{AdaptiveMaxPool}_{h_4,w_4}(\mathrm{Reshape}_{h_u,w_u}(U_L))\right] \tag{5}$$

$$\alpha_j = \frac{\exp(\mathrm{GAP}(\mathrm{Gate}_j(Z_{\mathrm{gl}})))}{\sum_{r=1}^{4} \exp(e_r)}, j = 1, \ldots, 4 \tag{6}$$

$$\boldsymbol{Z}_j = \alpha_j \odot Z_{\mathrm{gl}}, \quad j = 1, \ldots, 4 \tag{7}$$

Each axial operation applies self-attention independently to the sequences along its specified spatial axis and projects the output back to c_4 channels. The adaptive max-pooling output size is fixed to (h_4, w_4), and all gating convolutions use stride 1.

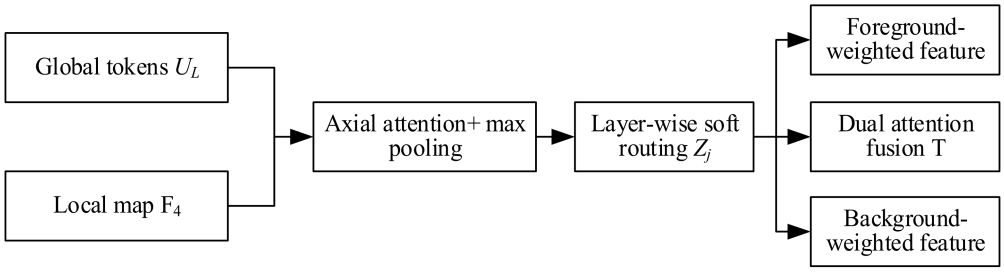


Figure 4. Detailed structure of the global optimization and foreground–background fusion attention module.

(3) Fusion Attention Module

The fusion attention module forms complementary foreground- and background-weighted representations at each fusion stage. First, the global representation Z_j is bilinearly resized to $h_j \times w_j$ and projected by a stage-dependent 1×1 convolution from c_{gl} channels to c_j channels. This operation is denoted by $\widetilde{Z}_j = \mathcal{R}_j(Z_j)$, giving $\widetilde{Z}_j \in \mathbb{R}^{c_j \times h_j \times w_j}$, which has the same shape as F_j. Equation (8) concatenates these features along the channel

dimension and uses a 1×1 convolution with one output channel, followed by a sigmoid, to produce the spatial foreground mask $B_j \in [0,1]^{1\times h_j \times w_j}$. In Equations (9) and (10), B_j and its complement $1 - B_j$ are broadcast over the feature channels to form foreground- and background-weighted features, respectively. Equations (11) and (12) aggregate the two branches from stage 4 to stage 1, with the recursion evaluated in the order $j = 3, 2, 1$. The operator $\mathcal{U}_j$ first bilinearly resizes a feature from stage $j + 1$ to $h_j \times w_j$ and then applies a 1×1 convolution from c_{j+1} channels to c_j channels. Its parameters are shared between the foreground and background branches at the same stage but differ across stages. All 1×1 convolutions use stride 1, and all bilinear interpolation uses `align_corners = False`. Finally, Equation (13) transforms the difference between the aggregated foreground and background features into the output representation T.

$$\widetilde{Z}_j = \mathcal{R}_j(Z_j), \mathbf{B}_j = \sigma(\mathrm{Conv}_{1\times1}(\mathrm{Cat}[\mathbf{F}_j, \mathcal{R}_j(\mathbf{Z_j})])), j = 1,2,3\ldots4 \tag{8}$$

$$F_j^{\mathrm{fg}} = B_j \odot F_j + B_j \odot \widetilde{Z}_j \tag{9}$$

$$F_j^{\mathrm{bg}} = (1 - B_j) \odot F_j + (1 - B_j) \odot \widetilde{Z}_j \tag{10}$$

$$G_4^{\mathrm{fg}} = F_4^{\mathrm{fg}}, \qquad G_j^{\mathrm{fg}} = F_j^{\mathrm{fg}} + \mathcal{U}_j(G_{j+1}^{\mathrm{fg}}), \quad j = 1,2,3 \tag{11}$$

$$G_4^{\mathrm{bg}} = F_4^{\mathrm{bg}}, \qquad G_j^{\mathrm{bg}} = F_j^{\mathrm{bg}} + \mathcal{U}_j(G_{j+1}^{\mathrm{bg}}), \quad j = 1,2,3 \tag{12}$$

$$T = \mathrm{PReLU}[\mathrm{Conv}_{3\times3}(|G_1^{\mathrm{fg}} - G_1^{\mathrm{bg}}|)] \tag{13}$$

3.2. UAV Recognition Algorithm Based on Cross-Axis Transformer Model

The AeroGRT cross-axis Transformer model includes a cross-axis attention mechanism, a query-reliability mask attention mechanism, a prediction head, and a training function. The AeroGRT model is more suitable for UAV detection and monitoring. Template–search feature interaction can be generalized to interactions between scene labeling, multi-scale feature maps, and detection queries.

(1) Cross-Axis Attention Mechanism

The cross-axis attention mechanism is shown in Figure 5. Equation (14) computes scaled dot-product attention for each head, and Equation (15) concatenates the head outputs. Here, n_{head} is the number of attention heads, d_{head} is the feature dimension of each head, and r indexes the heads. Equations (16) and (17) denote the attention and feed-forward substeps of an encoder layer, respectively, with $\tilde{Z}{}_{\mathrm{enc}}^{(\ell)}$ denoting the intermediate representation. The directional transformation block uses a fixed working grid of size $h_y \times w_y$. Before directional transformation, an incoming feature with a different grid is bilinearly resized to this working grid using `align_corners = False`. The resulting input is denoted by $Y_{\mathrm{ori}} \in \mathbb{R}^{c_y \times h_y \times w_y}$, where $c_y = n_{\mathrm{head}} d_{\mathrm{head}}$. Equation (18) applies horizontal and vertical transformations separately to each channel, with $W_{\mathrm{hor}} \in \mathbb{R}^{w_y \times w_y}$ and $W_{\mathrm{ver}} \in \mathbb{R}^{h_y \times h_y}$. Equation (19) concatenates the outputs along the channel dimension and uses a 1×1 convolution, denoted by FCN, to map $2c_y$ channels back to c_y channels while preserving the working grid. Equation (20) flattens the spatial dimensions into token sequences, uses the original sequence as the query and the fused sequence as the key–value source, and reshapes the attention output back to the working grid.

$$O_r^{\mathrm{head}} = \mathrm{Softmax}(\frac{Q_r K_r^{\top}}{\sqrt{d_{\mathrm{head}}}}) V_r, \quad r = 1, \ldots, n_{head} \tag{14}$$

$$\mathrm{MHAM}(Q,K,V) = \mathrm{Concat}(O_1^{\mathrm{head}},\ldots,O_{n_{\mathrm{head}}}^{\mathrm{head}})W^O \tag{15}$$

$$\tilde{Z}_{\mathrm{enc}}^{(\ell)} = \mathrm{LN}[Z_{\mathrm{enc}}^{(\ell-1)} + \mathrm{MHAM}(Z_{\mathrm{enc}}^{(\ell-1)}, Z_{\mathrm{enc}}^{(\ell-1)}, Z_{\mathrm{enc}}^{(\ell-1)})] \tag{16}$$

$$Z_{\mathrm{enc}}^{(\ell)} = \mathrm{LN}[\tilde{Z}_{\mathrm{enc}}^{(\ell)} + \mathrm{FFN}(\tilde{Z}_{\mathrm{enc}}^{(\ell)})] \tag{17}$$

$$\mathbf{X}_{hor} = \mathbf{W}_{hor}\mathbf{Y}_{\mathrm{ori}}, \qquad \mathbf{X}_{ver} = \mathbf{Y}_{\mathrm{ori}}\mathbf{W}_{ver} \tag{18}$$

$$\mathbf{Y}_{\mathrm{fusion}} = \mathrm{FCN}\ (\mathrm{Cat}[\mathbf{X}_{hor},\mathbf{X}_{ver}]) \tag{19}$$

$$\mathrm{CAM}(\mathbf{Y}_{\mathrm{ori}}) = \mathrm{MHAM}(\mathbf{Y}_{\mathrm{ori}}\mathbf{W}^Q,\mathbf{Y}_{\mathrm{fusion}}\mathbf{W}^K,\mathbf{Y}_{\mathrm{fusion}}\mathbf{W}^V) \tag{20}$$

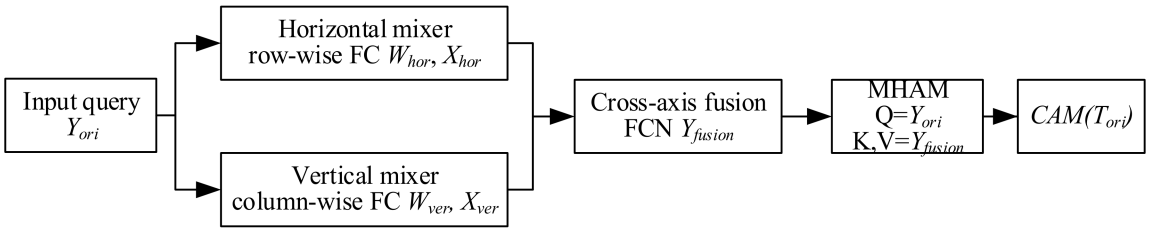


Figure 5. Specific details of the cross-axis attention mechanism for UAV image recognition.

(2) Query-Reliability Mask Attention

The query-reliability graph attention module is shown in Figure 6. In complex UAV scenes, foreground and background tokens can have similar features, resulting in ambiguous query–key associations. Equations (21)–(24) describe the computation for one attention head. Let $\mathbf{q}_i, \mathbf{k}_k \in \mathbb{R}^{d_{\mathrm{head}}}$ denote the query and key feature vectors, respectively, where $i = 1, \ldots, N_q$ and $k = 1, \ldots, N_k$. Equation (21) defines their similarity scores and the strongest unperturbed query correspondence i_k^* for each key. The index i_k^* explains the hard correspondence, while differentiable matching uses the distribution in Equation (22). The correspondence probabilities ϕ_{ki} are obtained by normalizing the similarity scores over queries for each key. Equation (22) applies a Gumbel–Softmax relaxation and computes the weighted query feature $\boldsymbol{\mu}_k$. During training, the noise variables ξ_i are independently sampled from the standard Gumbel distribution and shared across keys; during inference, $\xi_i = 0$. The temperature $\tau > 0$ controls the softness of the correspondence distribution. Equation (23) constructs a feature-space consistency mask based on the distance between each query feature and $\boldsymbol{\mu}_k$, where $\lambda > 0$ controls the mask bandwidth. Equation (24) incorporates this mask into query-to-key attention before aggregating the value vectors $\mathbf{v}_k$. The correspondence distribution in Equation (22) is normalized over queries for each key, whereas the attention weights in Equation (24) are normalized over keys for each query. This formulation favors feature-consistent query–key associations without assigning a spatial interpretation to feature distances.

$$i_k^* = \underset{1\le i\le N_q}{\mathrm{argmax}}\ a_{kq}, \qquad a_{ki} = \frac{\mathbf{q}_i^\top \mathbf{k}_k}{\sqrt{d_{\mathrm{head}}}} \tag{21}$$

$$\pi_{ki} = \frac{\exp[(a_{ki}+\xi_i)/\tau]}{\sum_{r=1}^{N_q}\exp[(a_{kr}+\xi_r)/\tau]}, \qquad \boldsymbol{\mu}_k = \sum_{i=1}^{N_q}\pi_{ki}\mathbf{q}_i \tag{22}$$

$$N_{ki} = \exp\left(-\frac{\|\mathbf{q}_i - \boldsymbol{\mu}_k\|_2^2}{\lambda^2}\right) \tag{23}$$

$$w_{ik} = \frac{\exp(a_{ki})N_{ki}}{\sum_{r=1}^{N_k}\exp(a_{ri})N_{ri}}, \mathbf{o}_i = \sum_{k=1}^{N_k} w_{ik}\mathbf{v}_k \tag{24}$$

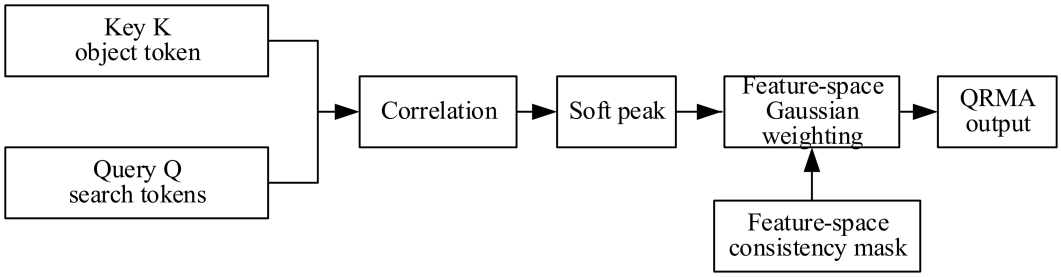


Figure 6. Flowchart of the query-reliability mask attention mechanism and its feature-space consistency mask.

(3) Scale–Altitude Routing Detection Head

The scale–altitude routing detection head is shown in Figure 7. It estimates image-level priors over pyramid levels P_2–P_5. Equation (25) globally pools each pyramid feature to obtain a scale-specific descriptor. Equation (26) concatenates these descriptors with normalized altitude metadata and an altitude-availability flag, and a two-layer MLP predicts four routing weights using softmax. Scale-related information is inferred from the pyramid features, and no ground-truth bounding-box dimensions are used as routing inputs during either training or inference. Each pyramid level retains decoupled classification, regression, and objectness branches. Equation (27) applies the routing weight once to the joint detection confidence, while retaining the decoded bounding-box coordinates without scaling. Candidates from all four levels are concatenated, filtered by confidence, and processed using class-wise non-maximum suppression. The routing network is trained jointly with the detection branches through the routed classification confidence.

$$P_\ell \in \mathbb{R}^{c_\ell^P \times h_\ell^P \times w_\ell^P}, \mathbf{u}_\ell = \mathrm{GAP}(P_\ell) \in \mathbb{R}^{c_\ell^P}, \ell = 2,3,4,5 \tag{25}$$

$$m_t = \begin{cases} 1, & \text{altitude metadata are available,} \\ 0, & \text{otherwise,} \end{cases} \quad \tilde{a}_t^{\mathrm{alt}} = \begin{cases} \frac{a_t^{\mathrm{alt}} - \mu_a}{\sigma_a + \varepsilon}, & m_t = 1, \\ 0, & m_t = 0. \end{cases} \tag{26}$$

$$\begin{aligned} r_{\ell uc} &= s_{t,\ell} o_{\ell u} p_{\ell uc}, \\ \mathcal{Y}_t &= \mathrm{NMS}_{\delta_{\mathrm{nms}}}(\mathrm{Concat}_{\ell=2}^{5}\{(\hat{\mathbf{b}}_{\ell u}, c, r_{\ell uc}) : r_{\ell uc} \geq \delta_{\mathrm{score}}\}). \end{aligned} \tag{27}$$

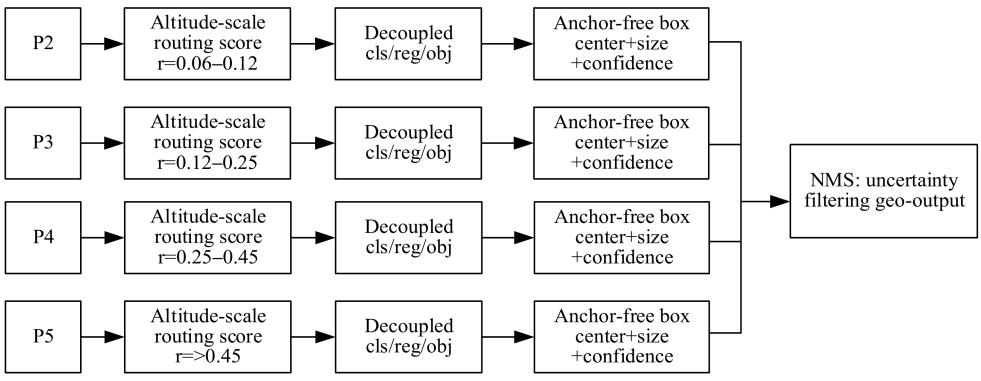


Figure 7. Scale–altitude routing detection head with decoupled classification, regression, and objectness branches.

Here, u indexes candidate detections, $c = 1, \ldots, n_{\mathrm{cls}}$ indexes object classes, $o_{\ell u}$ is the objectness probability, and $p_{\ell uc}$ is the conditional class probability. The decoded boxes $\hat{\mathbf{b}}_{\ell u}$ from all pyramid levels are expressed in the same image coordinate system. The thresholds δ_{score} and δ_{nms} control confidence filtering and class-wise NMS, respectively. Routing weights modify detection confidence only; they do not multiply bounding-box

coordinates. During training, label assignment and Equation (28) use dense predictions before confidence filtering and NMS. Confidence filtering and class-wise NMS are applied only during inference.

(4) Training Function

AeroGRT uses a multi-task objective consisting of classification loss, box regression loss, and query-consistency regularization. Let m index the valid prediction–class pairs after label assignment, let $y_m \in \{0, 1\}$ denote the corresponding binary label, and let r_m denote the routed joint confidence defined in Equation (27). The probability assigned to the observed label is $p_m^* = y_m r_m + (1 - y_m)(1 - r_m)$. Equation (28) applies focal loss to this probability, where ω_{y_m} is the class-balancing weight and $\gamma \geq 0$ is the focusing parameter. Equation (29) computes box regression loss over the N_{pos} positive matches, where $\hat{\mathbf{b}}_u$ and $\mathbf{b}_u^{\text{gt}}$ denote predicted and ground-truth boxes, respectively, and $\beta_1, \beta_2 \geq 0$ weight the L1 and GIoU terms. The box loss is zero when there are no positive matches. Equation (30) regularizes the agreement between the normalized feature-space consistency mask and the unmasked key-to-query correspondence distribution. For each key k, both distributions are normalized over queries, with $\overline{N}_{ki} = N_{ki} / \sum_{r=1}^{N_q} N_{kr}$ and $\overline{A}_{ki} = \phi_{ki}$. They differ from the query-to-key attention weights w_{ik} in Equation (24). The constant $0 < \varepsilon < 1$ provides numerical stability in the logarithms. Equation (31) combines the three losses, with $\beta_3 \geq 0$ controlling the contribution of query-consistency regularization.

$$p_m^\star = y_m r_m + (1 - y_m)(1 - r_m), \mathcal{L}_{\text{cls}} = -\frac{1}{N_{\text{samp}}} \sum_{m=1}^{N_{\text{samp}}} \omega_{y_m} (1 - p_m^\star)^\gamma \log(\max(p_m^\star, \varepsilon)) \tag{28}$$

$$\mathcal{L}_{\text{box}} = \frac{1}{N_{\text{pos}}} \sum_{u=1}^{N_{\text{pos}}} [\beta_1 \|\hat{\mathbf{b}}_u - \mathbf{b}_u^{\text{gt}}\|_1 + \beta_2 (1 - \text{GIoU}(\hat{\mathbf{b}}_u, \mathbf{b}_u^{\text{gt}}))] \tag{29}$$

$$\overline{N}_{ki} = \frac{N_{ki}}{\sum_{r=1}^{N_q} N_{kr}}, \overline{A}_{ki} = \phi_{ki} = \frac{\exp(a_{ki})}{\sum_{r=1}^{N_q} \exp(a_{kr})} \mathcal{L}_{\text{rel}} = \frac{1}{N_k} \sum_{k=1}^{N_k} \sum_{i=1}^{N_q} \overline{N}_{ki} \log\left(\frac{\overline{N}_{ki} + \varepsilon}{\overline{A}_{ki} + \varepsilon}\right) \tag{30}$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}} + \beta_3 \mathcal{L}_{\text{rel}} \tag{31}$$

The routing MLP receives gradients through the routed confidence in Equation (28), while box regression uses decoded boxes before confidence routing. The class-balancing weights satisfy $\boldsymbol{\omega_0}, \boldsymbol{\omega_1} \geq \mathbf{0}$. The classification loss is zero when there are no valid prediction–class pairs. The same routing computation is used during training and inference, with confidence filtering and NMS restricted to inference.

4. Experiments and Analysis

4.1. Environment Setting and Datasets

Table 2 summarizes the benchmark datasets considered for evaluating UAV recognition, tracking, and monitoring capabilities. These datasets cover single-object tracking, large-scale visual tracking, UAV-based object detection, small-object detection, and road defect recognition tasks. Among them, GOT-10k, TrackingNet, VisDrone2019, and TinyPerson are adopted for quantitative evaluation, while other datasets are introduced as representative benchmarks related to UAV perception and monitoring applications.

The GOT-10k dataset is adopted for single-object tracking evaluation. It contains 10,000 training sequences and 180 test sequences, covering more than 560 object categories and diverse motion patterns. Different from conventional tracking benchmarks, GOT-10k follows a strict one-shot evaluation protocol, where object categories in the test set do not overlap with those in the training set. Therefore, it provides a challenging benchmark for

evaluating the generalization ability of tracking models under unseen object categories and complex visual variations.

Table 2. Relevant datasets used for UAV recognition, tracking, and road monitoring evaluation.

Dataset	Scale and Split	Main Challenge
Got-10k	10,000 training sequences; 180 test sequences	Class diversity and strict one-shot protocol
TrackingNet	30,132 training sequences; 511 test sequences; about 14M boxes	Real-world YouTube videos with 15 attributes
UAV123	123 low-altitude UAV video sequences	Camera motion, altitude variation, scale change
VisDrone2019	6471 train; 548 val; 1610 test-dev; 10 object classes	Dense small objects in 14 cities
TinyPerson	1610 images, commonly split 8:1:1	Tiny pedestrians under high-altitude views
RDD2022-China	4878 China subset images	Ground and drone viewpoints; multiple damage types
UAV-PDD2023	2440 three-channel JPG images	Low contrast and small road damages under UAV imaging
GRDDC2020	21,041 images from Czech Republic, India, and Japan	Cross-country background diversity

The TrackingNet dataset is further used to evaluate large-scale tracking performance in real-world scenarios. It contains 30,132 training sequences and 511 test sequences collected from YouTube videos, with approximately 14 million annotated bounding boxes. It emphasizes tracking robustness under diverse appearance changes and complex environmental conditions.

For UAV-based object detection evaluation, the VisDrone2019 and TinyPerson datasets are selected. VisDrone2019 contains images captured from multiple urban environments with dense small objects, while TinyPerson focuses on tiny pedestrian detection from high-altitude views. These datasets provide challenging scenarios for evaluating UAV perception under scale variation, background interference, and low-resolution targets.

To provide a clearer understanding of the tracking task, representative samples from the GOT-10k dataset are presented in Figure 8. The examples illustrate target appearance variation, scale changes, background interference, and motion deformation.



Figure 8. Representative tracking samples from the GOT-10k dataset.

The detailed information of all benchmark datasets is summarized in Table 2.

For a complete AeroGRT implementation, the default training setting is PyTorch 2.14.1 with AdamW or SGD, an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}, 300 training epochs, a batch size of 8, mosaic and mixup augmentation for detection datasets, random shift and blur augmentation for tracking datasets, and multi-scale training for UAV object detection. We plan to explore ONNX model export and

TensorRT inference optimization, followed by measurements of end-to-end latency, frame rate, power consumption, and memory usage on Jetson-class embedded devices to assess deployment feasibility. The evaluation pipeline is shown in Figure 9. Equation (32) defines precision, Equation (33) defines recall, Equation (34) defines class-wise AP and the mean AP over all classes, Equation (35) gives the harmonic F1 measure, and Equation (36) reports inference throughput in frames per second.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{32}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{33}$$

$$\text{AP}_c = \int_0^1 P_c(R)\,\mathrm{d}R, \qquad \text{mAP} = \frac{1}{C}\sum_{c=1}^{C} \text{AP}_c \tag{34}$$

$$F_1 = 2\,\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{35}$$

$$\text{FPS} = \frac{N_{\text{frames}}}{T_{\text{inference}}} \tag{36}$$

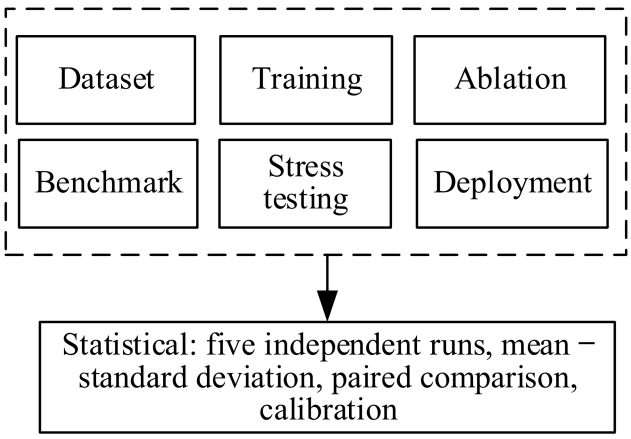


Figure 9. Pipeline for benchmark evaluation and planned edge deployment.

4.2. Ablation Experiments

Table 3 lists the ablation results on GOT-10k. IFENM, CAM, and RMA denote the interfusion feature extraction module, cross-axis attention module, and query-reliability mask attention module, respectively. The baseline has an AO of 63.4%. The individual IFENM, CAM, and RMA variants have AO values of 72.9%, 72.3%, and 69.5%, respectively. The IFENM + CAM, CAM + RMA, and IFENM + RMA combinations have AO values of 73.4%, 72.8%, and 74.1%, respectively. The complete combination has an AO of 75.8%, an SR0.5 of 83.9%, and an SR0.75 of 73.1%. Figure 10 visualizes the ablation results.

Table 4 summarizes the ablation trend reported by AnisoNet for geometry-aware recognition. The baseline reaches 86.7% mAP@50. Adding LKMSA increases mAP@50 by 2.5 percentage points while reducing FPS only moderately. MRDC raises mAP@50 to 90.5%, SRBlock reaches 91.8%, and the complete AnisoNet architecture reaches 92.8%. Figure 11 visualizes this mAP@50 trend. These results motivate the use of geometry-aware feature modeling in AeroGRT.

The performance results are reported as mean ± standard deviation (mean ± SD). A smaller standard deviation indicates that the model achieves more stable performance across different training runs. Furthermore, a paired two-tailed Student's *t*-test was performed between the baseline Transformer model and AeroGRT to determine whether

the observed improvements were statistically significant rather than caused by random fluctuations. The statistical comparison results on the GOT-10k dataset are presented in Table 5.

Table 3. GOT-10k ablation results (unit: %).

Sequence	IFENM	CAM	RMA	AO	SR0.5	SR0.75
1	-	-	-	63.4	74.9	51.4
2	√	-	-	72.9	81.1	71.3
3	-	√	-	72.3	80.2	70.8
4	-	-	√	69.5	76.7	66.4
5	√	√	-	73.4	81.9	72.1
6	-	√	√	72.8	80.9	71.1
7	√	-	√	74.1	82.1	71.9
AeroGRT	√	√	√	75.8	83.9	73.1

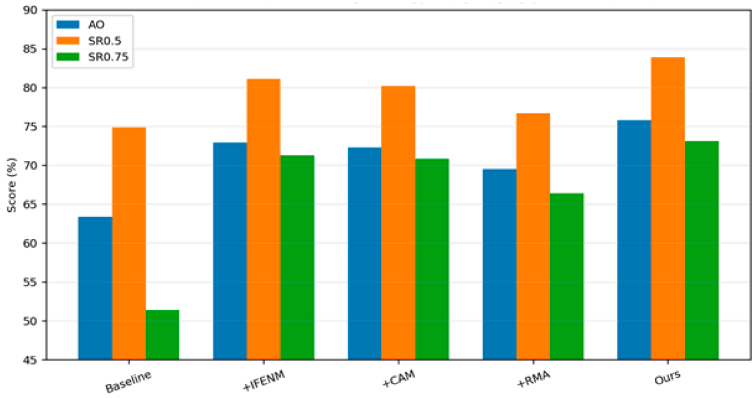


Figure 10. Ablation experiment results on the Got-10k dataset.

Table 4. AnisoNet ablation trend used as geometry-aware evidence.

Variant	Core Module	mAP@50 (%)	Efficiency Information	Relevance to AeroGRT
M1	Baseline	86.7	125.64 FPS	Base one-stage detector
M2	LKMSA	89.2	121.48 FPS	Multi-scale large-kernel enhancement
M3	MRDC	90.5	119.37 FPS	Multi-route dynamic semantic modeling
M5	SRBlock	91.8	118.25 FPS	Geometric and serpentine topology modeling
M8	Complete AnisoNet	92.8	115.49 FPS	Synergistic design of attention and geometry

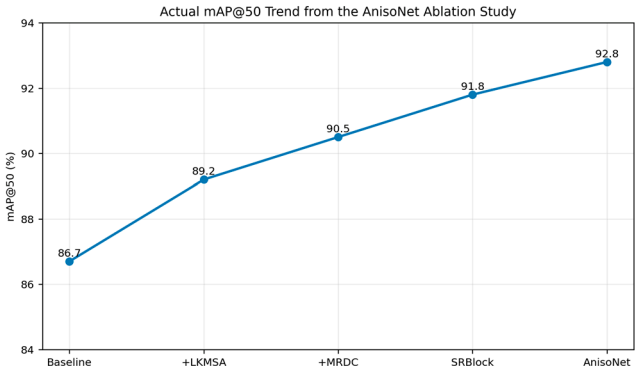


Figure 11. Plot of mAP@50 trend from the AnisoNet ablation study.

Compared with the baseline, AeroGRT improves AO, SR0.5, and SR0.75 by 12.3, 9.1, and 21.5 percentage points, respectively. The paired two-tailed Student's *t*-test indicates

statistically significant differences across all three metrics ($p < 0.001$), providing statistical evidence of improved performance across repeated training runs.

Table 5. Statistical significance analysis between the baseline and AeroGRT on GOT-10k.

Method	AO	SR0.5	SR0.75
Baseline	63.5 ± 0.42	74.8 ± 0.36	51.6 ± 0.51
AeroGRT	75.8 ± 0.31	83.9 ± 0.28	73.1 ± 0.35

4.3. Performance Evaluation of Traffic Target Tracking and UAV Recognition

(1) Experimental Results on the Got-10k Dataset

Table 6 presents the comparative tracking results on GOT-10k. The AeroGRT row reports an AO of 75.8%, an SR0.5 of 83.9%, and an SR0.75 of 73.1%, corresponding to gains of 1.0, 2.0, and 0.9 percentage points over the listed SeqTrack results. Its reported inference speed is 60.9 FPS, compared with 68.6 FPS for SimTrack. The table compares tracking accuracy and inference speed across the listed methods

Table 6. Comparative tracking results on GOT-10k (unit: %).

Method	AO	SR0.5	SR0.75	FPS
Baseline	63.4	74.9	51.4	54.5
SiamST	62.1	72.5	51.6	25.81
SimTrack	69.8	78.8	66.0	68.6
CSwinTT	69.4	78.9	65.4	12.4
SLT	67.5	76.5	60.3	50
AiATrack	69.6	80.0	63.2	38
MixFormer-22k	70.7	80.0	67.8	25
SeqTrack	74.8	81.9	72.2	5
AeroGRT	75.8	83.9	73.1	60.9

(2) Experiment Results on the TrackingNet Dataset

Table 7 reports the TrackingNet comparative results. The AeroGRT method achieves AUC of 87.4%, normalized precision of 91.3%, and precision of 87.6%. Compared with SeqTrack, the improvements are 1.9, 1.5, and 1.8 percentage points. Figure 12 plots these results and shows that the method improves both overlap-based and precision-based measures. These results demonstrate AeroGRT's improved overall tracking performance on TrackingNet.

Table 7. Comparison of tracking performance on TrackingNet (unit: %).

Method	AUC	Pnorm	*p*
Baseline	58.1	67.3	54.3
SiamST	65.2	70.9	62.1
SimTrack	83.4	87.4	-
CSwinTT	81.9	86.7	79.5
SLT	82.8	87.5	81.4
AiATrack	82.7	87.8	80.4
MixFormer-22k	83.1	88.1	81.6
SeqTrack	85.5	89.8	85.8
AeroGRT	87.4	91.3	87.6

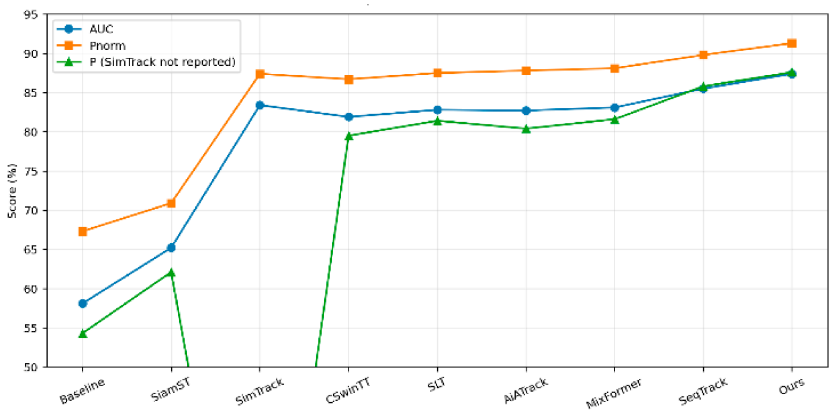


Figure 12. Comparative tracking results on TrackingNet.

(3) Small-Object Detection Results on VisDrone and TinyPerson

Table 8 presents AeroGRT's small-object detection improvements on VisDrone and TinyPerson. On VisDrone, AeroGRT improves mAP50 and mAP50–95 by 2.8 and 1.4 percentage points, respectively, over YOLOv8n + P2 + P3. On TinyPerson, the corresponding improvements over YOLOv8n + P2 + P3 + P4 are 1.1 and 0.2 percentage points. Compared with YOLOv10n + P2 + P3 + P4 + P5 on VisDrone, AeroGRT improves the two metrics by 3.4 and 1.4 percentage points, respectively.

Table 8. Small-object detection gains on VisDrone and TinyPerson.

Method	Dataset	Reference Comparison	mAP50 Gain	mAP50-95 Gain
AeroGRT	VisDrone	YOLOv8n + P2 + P3	+2.8	+1.4
AeroGRT	TinyPerson	YOLOv8n + P2 + P3 + P4	+1.1	+0.2
AeroGRT	VisDrone	YOLOv10n + P2 + P3 + P4 + P5	+3.4	+1.4

The experimental results reveal that different components of AeroGRT contribute to performance improvement from complementary perspectives. In the AeroGRT tracking experiments, inter-fusion feature extraction provides the largest individual improvement, indicating that global–local feature interaction is beneficial for handling appearance variation and background interference. Cross-attention further enhances localization accuracy by preserving boundary and contour information, while relevance mask attention reduces unreliable query–key associations.

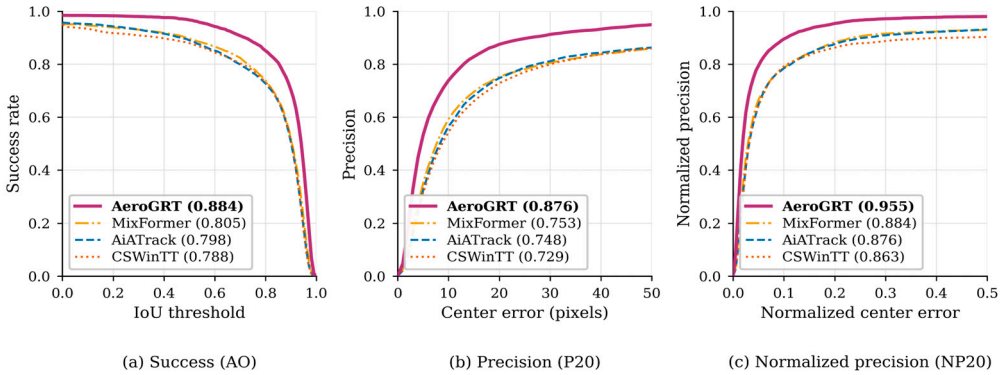
In addition, the geometry-aware mechanisms introduced from AnisoNet provide useful guidance for UAV recognition tasks involving slender, weak-texture, and low-contrast targets. The combination of global–local feature interaction and geometry-adaptive routing enables AeroGRT to better handle challenging UAV scenarios, including high-altitude views, cluttered urban environments, and infrastructure monitoring scenes.

However, the current evaluation mainly focuses on benchmark performance. Further work will address calibration evaluation, edge deployment latency, power consumption, and code availability.

4.4. Visualization Experiments

To further comprehensively verify the performance of the AeroGRT model, we conduct visualization experiments including quantitative curves, tracking outputs and corresponding attention maps. We compare four pre-trained trackers on a fixed subset of 50 randomly sampled sequences from the AG validation set. All trackers are initialized with the ground-truth bounding box in the first frame, without subsequent ground-truth re-initialization.

Metrics are computed over the identical set of 5655 valid non-initialization frames and equally averaged across sequences. The results are illustrated in Figures 13–15.



(a) Success (AO) (b) Precision (P20) (c) Normalized precision (NP20)

Figure 13. Overall performance. (**a**) Success plot, (**b**) center location precision plot, (**c**) normalized center location precision plot. All curves are calculated per sequence and equally averaged across 50 sequences.

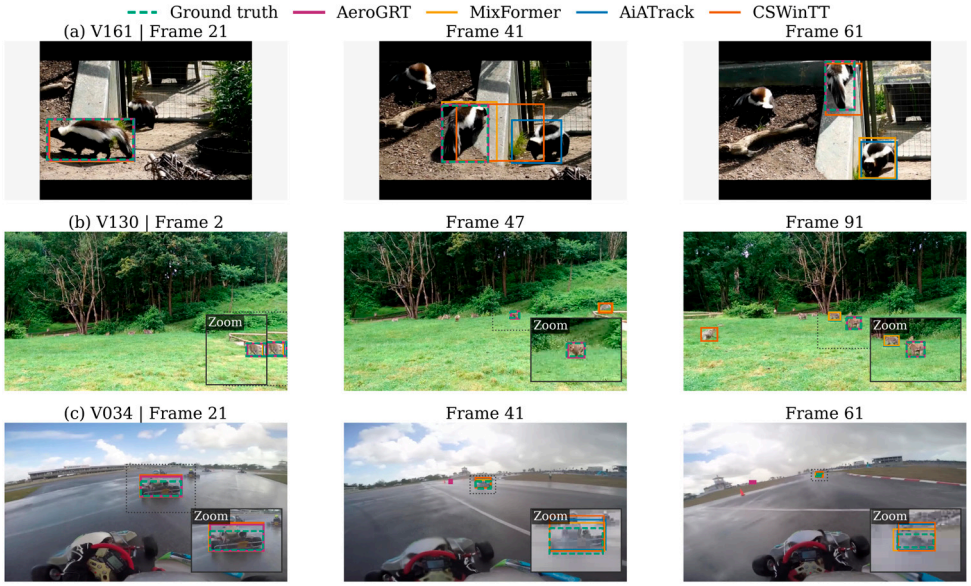


Figure 14. Qualitative tracking results. Green dashed lines are ground truth, purple solid lines are AeroGRT, and other colors denote baselines. The three rows show similar-object interference, small-target tracking, and a failure case.

As shown in Figure 13, AeroGRT achieves the best overall localization performance, with an AO of 0.8836, a P20 of 87.56%, and an NP20 of 95.45%. Compared with the strongest baseline, MixFormer, AeroGRT improves AO by 0.0783 and P20 and NP20 by 12.25 and 7.08 percentage points, respectively. The 95% confidence interval for the AO improvement, estimated through paired sequence-level bootstrap resampling, is [0.043, 0.119], supporting a consistent advantage on the evaluated subset.

Figure 14 presents frame-level qualitative results consistent with the quantitative findings. Sequence V161 contains visually similar distractors that challenge target identity preservation. At frame 41, AeroGRT achieves an IoU of 0.970, compared with 0.774, 0.301, and 0 for MixFormer, CSWinTT, and AiATrack, respectively. By frame 61, AeroGRT maintains an IoU of 0.965, whereas AiATrack and MixFormer have drifted away from the annotated target. In V130, AeroGRT successfully localizes the small target at frame 91 with an IoU of 0.922, while all three baselines exhibit zero overlap with the ground truth. These examples illustrate more reliable target retention under similar-instance interference and

small-target conditions. Figure 14c also reveals a failure case: as the go-kart recedes in V034, AeroGRT loses the target by frame 41, while MixFormer retains an IoU of 0.714. This case highlights that the aggregate performance advantage does not guarantee reliable tracking across all challenging sequences.

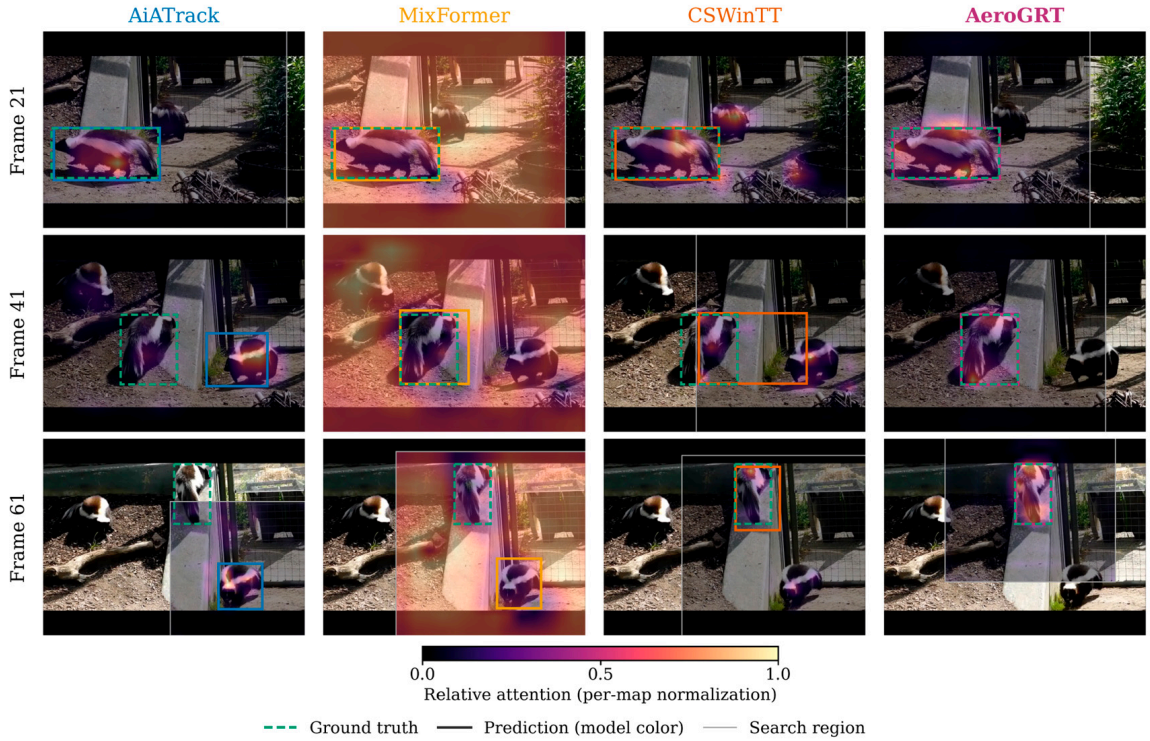


Figure 15. Corresponding attention maps. Brighter regions indicate higher relative attention within each map. Gray borders denote the search region.

The attention maps in Figure 15 provide a qualitative interpretation of the tracking behavior in V161. At frames 41 and 61, AeroGRT's high-response regions remain close to the ground-truth target, spatially consistent with its accurate predictions. In contrast, some baseline maps highlight distractor instances or broader background regions. This spatial agreement between attention responses and localization results is consistent with AeroGRT's stronger target retention in the illustrated sequence.

5. Conclusions

This paper presents AeroGRT, a geometry-aware global–local routing Transformer with query-reliability graphs for UAV visual perception. The formulation combines complementary global–local feature extraction, scalar stage weighting of a shared global representation, and foreground–background feature fusion. Cross-axis attention models directional contextual information, while a feature-space consistency mask weights query–key associations. A pyramid-level routing head uses pooled image features and optional altitude metadata to modulate detection confidence without scaling box coordinates. These components are designed to address local-detail preservation, boundary-sensitive representation, feature association, and scale variation in complex UAV scenes.

AeroGRT was evaluated for object tracking on GOT-10k and TrackingNet and for small-object detection on VisDrone and TinyPerson. The GOT-10k ablation results show that IFENM, CAM, and RMA each improve AO over the baseline, and their full combination achieves the highest AO among the tested variants. Visualization experiments on 50 AG validation sequences further supplement the above analyses via performance curves, frame-

wise tracking comparisons and attention maps. The examples demonstrate more robust target retention under similar-object interference and small-target scenarios, with attention responses well aligned to annotated targets. Future work will focus on broader multi-dataset evaluation, computational-efficiency optimization, and real-time deployment on UAV edge-computing platforms.

Author Contributions: Conceptualization, Y.Z.; Methodology, Y.Z.; Software, X.M.; Validation, X.M.; Formal Analysis, Z.L.; Investigation, X.M.; Resources, Z.L.; Data Curation, X.M.; Writing—Original Draft Preparation, Y.Z.; Writing—Review and Editing, X.M.; Visualization, M.M.; Supervision, Z.L.; Project Administration, M.M.; Funding Acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by China Postdoctoral Science Foundation funded project (No. 2021M692400), Yunnan Key Laboratory of Unmanned Autonomous Systems (No. 202502YB04), Open Research Fund of State Key Laboratory of Integrated Services Networks (Xidian University, No. ISN24-22).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are not publicly available due to privacy and institutional restrictions. The data may be available from the corresponding author upon reasonable request and with permission of the data provider.

Conflicts of Interest: Yihua Zang, Xingmin Ma was employed by the China Electronics Technology Group Corporation. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7380–7399. [CrossRef]
2. Yu, H.; Li, G.; Zhang, W.; Huang, Q.; Du, D.; Tian, Q.; Sebe, N. The unmanned aerial vehicle benchmark: Object detection, tracking and baseline. *Int. J. Comput. Vis.* **2020**, *128*, 1141–1159. [CrossRef]
3. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for UAV tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 445–461. [CrossRef]
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
6. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
7. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [CrossRef]
8. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision (ECCV), Milan, Italy, 29 September–4 October 2024; pp. 1–21. [CrossRef]
9. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458. [CrossRef]
10. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs beat YOLOs on real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974. [CrossRef]

11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
12. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
13. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training data-efficient image transformers and distillation through attention. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 10347–10357.
14. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 213–229. [CrossRef]
15. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 548–558. [CrossRef]
16. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 9992–10002. [CrossRef]
17. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–14 December 2021; pp. 12077–12090.
18. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8126–8135. [CrossRef]
19. Ai, D.; Jiang, G.; Lam, S.-K.; He, P.; Li, C. Computer vision framework for crack detection of civil infrastructure: A review. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105478. [CrossRef]
20. Qiu, Q.; Lau, D. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to UAV images. *Autom. Constr.* **2023**, *147*, 104745. [CrossRef]
21. He, X.; Tang, Z.; Deng, Y.; Zhou, G.; Wang, Y.; Li, L. UAV-based road crack object-detection algorithm. *Autom. Constr.* **2023**, *154*, 105014. [CrossRef]
22. Li, C.; Fan, Z.; Chen, Y.; Lin, H.; Moretti, L.; Loprencipe, G.; Sheng, W.; Wang, K.C.P. CrackCLF: Automatic pavement crack detection based on closed-loop feedback. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 5965–5980. [CrossRef]
23. Zhu, Z.; Huang, D.; Zhou, X.; Chen, D.; Fu, J.; Yang, J. Image quality evaluation method for surface crack detection based on standard test chart. *Comput.-Aided Civ. Infrastruct. Eng.* **2024**, *39*, 1294–1317. [CrossRef]
24. Wu, Y.; Hong, M.; Li, A.; Huang, S.; Liu, H.; Ge, Y. Self-supervised adversarial learning for domain adaptation of pavement distress classification. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 1966–1977. [CrossRef]
25. Arya, D.; Maeda, H.; Ghosh, S.K.; Toshniwal, D.; Sekimoto, Y. RDD2022: A multi-national image dataset for automatic road damage detection. *Geosci. Data J.* **2024**, *11*, 846–862. [CrossRef]
26. Yan, H.; Zhang, J. UAV-PDD2023: A benchmark dataset for pavement distress detection based on UAV images. *Data Brief* **2023**, *51*, 109692. [CrossRef]
27. Huo, J.; Wu, Z.; Kong, C. AnisoNet: A Detecting Model for Initial Anisotropic Cracks in Complex Road Scenarios. *IEEE Trans. Intell. Transp. Syst.* **2026**, *27*, 12247–12264. [CrossRef]
28. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R.W. BiFormer: Vision transformer with bi-level routing attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 10323–10333. [CrossRef]
29. Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; Yang, G. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 6070–6079. [CrossRef]
30. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31 × 31: Revisiting large kernel design in CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11963–11975. [CrossRef]
31. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-Style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
32. Guo, M.-H.; Lu, C.-Z.; Hou, Q.; Liu, Z.; Cheng, M.-M.; Hu, S.-M. SegNeXt: Rethinking convolutional attention design for semantic segmentation. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; pp. 1140–1156.

33. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning spatio-temporal transformer for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10448–10457. [CrossRef]
34. Chen, X.; Peng, H.; Wang, D.; Lu, H.; Hu, H. SeqTrack: Sequence to sequence learning for visual object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 14572–14581. [CrossRef]
35. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. MixFormer: End-to-end tracking with iterative mixed attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 13608–13618. [CrossRef]
36. Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; Ling, H. SwinTrack: A simple and strong baseline for transformer tracking. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; pp. 16743–16754.
37. Gao, S.; Zhou, C.; Ma, C.; Wang, X.; Yuan, J. AiATrack: Attention in attention for transformer visual tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 146–164. [CrossRef]
38. Xing, D.; Evangeliou, N.; Tsoukalas, A.; Tzes, A. Siamese transformer pyramid networks for real-time UAV tracking. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022; pp. 2139–2148. [CrossRef]
39. Song, Z.; Luo, R.; Yu, J.; Chen, Y.P.P.; Yang, W. Compact transformer tracker with correlative masked modeling. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 2321–2329. [CrossRef]
40. Fu, C.; Lu, K.; Zheng, G.; Ye, J.; Cao, Z.; Li, B.; Lu, G. Siamese object tracking for unmanned aerial vehicle: A review and comprehensive analysis. *Artif. Intell. Rev.* **2023**, *56*, 1417–1477. [CrossRef]
41. Liao, X.; Guo, X.; Rozi, A.; Yu, H.; Haji, A. End to end polysemantic cooperative mixed task trainer for UAV target detection. *Sci. Rep.* **2024**, *14*, 29775. [CrossRef]
42. Ou, K.; Dong, C.; Liu, X.; Zhai, Y.; Li, Y.; Huang, W.; Qiu, W.; Wang, Y.; Wang, C. Drone-TOOD: A lightweight task-aligned object detection algorithm for vehicle detection in UAV images. *IEEE Access* **2024**, *12*, 41999–42016. [CrossRef]
43. Bai, J.; Hu, H.; Li, Y.; Liu, X.; Zhuang, S.; Wang, Z. UAV image object detection based on self-attention guidance and global feature fusion. *Image Vis. Comput.* **2024**, *151*, 105262. [CrossRef]
44. Liu, J.; Xie, Y. WDFS-DETR: A transformer-based framework with multi-scale attention for small object detection in UAV engineering tasks. *Results Eng.* **2025**, *27*, 105930. [CrossRef]
45. Zunair, H.; Khan, S.; Ben Hamza, A. RSUD20K: A Dataset for Road Scene Understanding in Autonomous Driving. In Proceedings of the 2024 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 27–30 October 2024; pp. 708–714. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.