SOAR: Scene-Aware Adaptive Resolution for Efficient UAV Multi-Object Tracking

Xujiang Tang[1,†], Lingying Zhao[1,†], Guangxi Chen[1,2], Jirong Huang[1] and Yutao Tang[1,2,*]
[1]Guilin University of Electronic Technology, Guilin, China
[2]Guangxi Academy of Artificial Intelligence, Nanning, China
Email: Yutao.Tang@guet.edu.cn

***Abstract*—Multi-object tracking (MOT) on unmanned aerial vehicle (UAV) platforms faces a fundamental tension between the need for rich visual detail and strict onboard computation constraints, with input resolution serving as a key lever in this trade-off. Existing methods either compress network complexity without addressing input redundancy, or adopt a fixed resolution that assumes scene homogeneity. However, we observe that resolution sensitivity varies markedly with scene conditions: according to perspective projection, target pixel scale is inversely proportional to observation distance, suggesting an optimal resolution matched to the current scene state. Our pilot study on the VisDrone and UAVDT (UAV Detection and Tracking) benchmarks confirms this hypothesis, showing that close-range sequences exhibit less than 1% performance gap across resolutions while distant sequences show gaps of 5%–11%, revealing systematic resolution redundancy in fixed-resolution pipelines. Based on these observations, we propose Scene-Aware Adaptive Resolution (SOAR), a framework that reconceptualizes input resolution from a static hyperparameter into a learnable decision variable by formulating resolution selection as a sequential decision problem and adopting reinforcement learning to model long-term tracking impact. SOAR infers scene state from proxy features such as detection confidence without requiring explicit ranging, and achieves 11%–21% computational savings with Multiple Object Tracking Accuracy (MOTA) drops under 1.2 points and Identification F1 (IDF1) drops under 0.8 points, or 20%–44% savings with MOTA drops under 2.5 points while IDF1 remains nearly unchanged.**

***Index Terms*—Multi-object tracking, unmanned aerial vehicle, adaptive resolution, reinforcement learning, computational efficiency**

I. Introduction

Multi-object tracking (MOT) is a core capability for visual perception on Unmanned Aerial Vehicles (UAVs), supporting applications such as surveillance, traffic monitoring, and search-and-rescue. However, UAV platforms face a fundamental tension in that high-quality tracking relies on rich visual detail, yet onboard computation and battery constraints impose strict limits on affordable workload. This tension between accuracy and efficiency becomes particularly acute in typical UAV scenarios involving small-object detection and long-range imaging, severely constraining the deployment of high-performance MOT systems in real-world missions.

Among the factors that influence this trade-off, *input resolution* plays a central role. Higher resolution preserves fine-grained texture and improves perception of small targets, yet computational cost grows quadratically with resolution. Prior work addresses this trade-off along two directions. One direction reduces network complexity through pruning, knowledge distillation, and related techniques [1], [2]. The other direction selects a fixed compromise resolution through offline profiling. However, the former operates only at the network-weight level and does not touch redundancy in the input data itself, while the latter implicitly assumes scene homogeneity and cannot adapt to the rapidly changing imaging conditions and target distributions typical of aerial perspectives. Recently, dynamic resolution selection has attracted attention as a new dimension for efficiency optimization [3], [4], but existing work primarily targets single-frame tasks such as image classification. These methods do not transfer directly to MOT, where resolution decisions affect not only current-frame detection quality but also future tracking states through the association mechanism.

The motivation of this work stems from a key observation about UAV imaging. Unlike fixed surveillance cameras, UAVs are in continuous motion during missions, causing the camera-to-target distance to vary dynamically over time. According to perspective projection, the pixel scale of a target is inversely proportional to the observation distance. When the UAV approaches a target, the target occupies more pixels and the system becomes less dependent on high resolution; conversely, higher resolution is needed to maintain feature discriminability. This regularity suggests the existence of an *optimal resolution* matched to the current scene state, one that satisfies tracking accuracy requirements without introducing unnecessary computational redundancy.

To validate this hypothesis, we conduct a pilot study on VisDrone [5] and UAVDT [6], two widely used UAV MOT benchmarks. The results reveal that resolution sensitivity varies substantially across scenes (Fig. 1). In close-range sequences where targets are large, the performance gap between the highest and lowest resolutions is less than 1%. In distant sequences where targets are small, the gap reaches 5%–11%. This contrast directly confirms the optimal-resolution hypothesis and exposes a systematic *resolution redundancy* problem in fixed-resolution pipelines. Section III-A further analyzes the cause of this phenomenon from the perspective

[†]Equal contribution.
[*]Corresponding author.

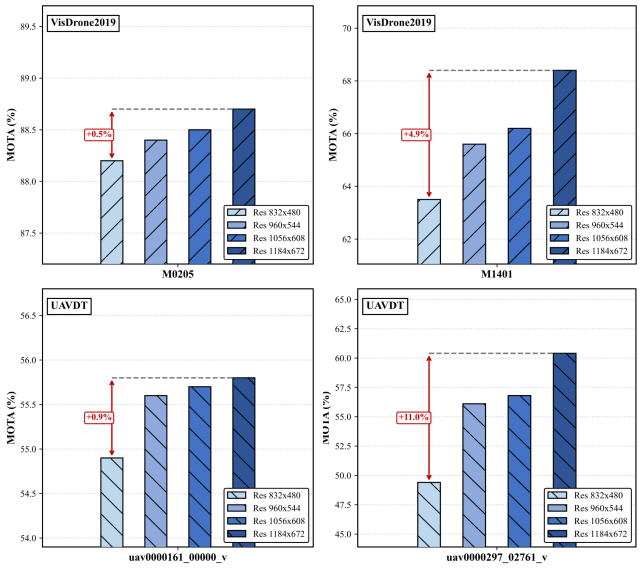

Fig. 1. MOTA performance comparison across different resolution levels for various sequences. Close-range sequences on the left exhibit low sensitivity to resolution, while distant sequences on the right are more sensitive.

of feature responses.

Based on these observations, we propose **Scene-Aware Adaptive Resolution (SOAR)**, a framework that reconceptualizes input resolution from a static hyperparameter into a learnable dynamic decision variable. We formulate resolution selection as a sequential decision problem and adopt reinforcement learning to model the long-term impact of current decisions on future tracking states. SOAR employs a modular design in which a lightweight policy network dynamically selects the optimal resolution based on scene state, and can be flexibly integrated with any detector and association algorithm. Since camera-to-target distance is difficult to obtain in real time, we design the state representation based on the principle of *indirect perception*, inferring scene state from features highly correlated with target scale, such as detection confidence, without requiring explicit ranging. Experiments on VisDrone and UAVDT demonstrate that SOAR achieves 11%–21% computational savings with Multiple Object Tracking Accuracy (MOTA) drops under 1.2 points and Identification F1 (IDF1) drops under 0.8 points. In a more aggressive configuration, SOAR attains 20%–44% savings with MOTA drops under 2.5 points while IDF1 remains nearly unchanged, indicating that identity consistency is well preserved under dynamic resolution adaptation. These results suggest that SOAR provides a practical and efficient solution for resource-constrained UAV perception systems.

This paper makes three contributions. First, we propose an *adaptive resolution scheduling framework* that reconceptualizes input resolution as a learnable sequential decision variable for temporal MOT tasks, using reinforcement learning to model the long-term impact of resolution choices on trajectory association, thereby shifting from static hyperparameter tuning to dynamic scene-adaptive scheduling. Second, we introduce *indirect-perception-based decision modeling* by formulating resolution selection as a Markov decision process with a state representation based on proxy features such as detection confidence rather than explicit ranging, together with a reward mechanism that jointly optimizes accuracy and efficiency. Third, we provide *systematic empirical validation* through comprehensive evaluation and ablation analysis on two mainstream benchmarks, demonstrating that dynamic resolution scheduling achieves substantial computational savings while preserving both MOTA and IDF1.

II. Related Work

A. Multi-Object Tracking on UAV Platforms

Multi-object tracking (MOT) on UAV platforms has received increasing attention due to its applications in surveillance, traffic monitoring, and search-and-rescue [5], [6]. Many approaches follow a tracking-by-detection pipeline where a detector localizes objects and an association module links detections across frames. ByteTrack [7] improves association by using low-confidence detections, while BoTSORT [8] incorporates camera motion compensation and Kalman filtering. Recent UAV trackers address challenges such as small-object scale [9], rapid viewpoint changes [10], and limited onboard compute [11]. Our work is orthogonal to tracker design and focuses on adaptive input resolution for efficiency.

B. Adaptive Resolution for Video Analytics

Efficiency optimization for edge video analytics includes model compression [1], [2] and dynamic networks, but these operate at the network level and do not directly address input redundancy. Dynamic resolution selection provides an alternative. Resolution Adaptive Networks [3] select resolution per image for classification, and Glance-and-Focus [4] uses coarse-to-fine processing. For video analytics, Chameleon [12] profiles configuration options offline. More recent systems adapt during deployment. DeepScale [13] uses online reinforcement learning to adjust frame size based on detection confidence, and FastTuner [14] predicts configuration combinations with a lightweight network. These methods often target fixed surveillance scenarios and do not explicitly model how resolution decisions affect later trajectory association in tracking. A missed detection at low resolution can lead to fragmentation even after resolution returns to a higher value. Online reinforcement learning can also degrade tracking during exploration.

C. Offline Reinforcement Learning for Resolution Scheduling

The temporal dependency in MOT motivates long-term reward modeling rather than per-frame heuristics. Offline reinforcement learning learns a policy from pre-collected data without online interaction and avoids failures during exploration. Conservative Q-Learning (CQL) [15] mitigates value overestimation in offline settings by penalizing out-of-distribution actions. We adopt CQL to learn a window-level resolution policy from offline trajectories and optimize the accuracy–efficiency trade-off over time.

Fig. 2. Feature response visualization across resolutions. Columns show resolutions from 832×480 to 1184×672. Panel (a) shows an easy scene with large objects where responses remain strong across resolutions. Panel (b) shows a challenging scene with small objects where low resolution weakens responses and higher resolution restores discriminative features.

III. Method

A. Problem Formulation

We study adaptive resolution scheduling for multi-object tracking (MOT) on UAV platforms. Given a video sequence $\{I_1, I_2, \ldots, I_T\}$, we partition it into non-overlapping temporal windows of length W frames. At each window t, the policy selects a discrete action $a_t \in \{0, 1, 2, 3\}$ that maps to an input resolution $r(a_t) \in \mathcal{R}$, and all frames within the window use this resolution. As shown in Sec. I, different scenes exhibit markedly different sensitivity to resolution: near-field sequences show MOTA variation of less than 1% across resolutions, while far-field sequences vary from 5% to 11%. Fig. 2 illustrates the underlying mechanism—in easy scenes with large objects, feature responses remain strong even at low resolution, whereas challenging scenes with small or distant targets require higher resolution to preserve discriminative features.

This scene-dependent behavior motivates formulating resolution scheduling as a Markov Decision Process (MDP). The MDP formulation captures the sequential dependency of tracking: the resolution chosen for the current window affects not only detection quality but also later tracking states through the association mechanism—if a target is missed at low resolution, the trajectory can fragment and become difficult to recover. Since we lack direct annotations of optimal resolution per window, we construct offline reference actions (Sec. III-D) and learn the accuracy–efficiency trade-off through long-term rewards. Formally, the MDP is $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is a 9-dimensional scene-aware state space (Sec. III-C), $\mathcal{A} = \{0, 1, 2, 3\}$ maps to the resolution set $\mathcal{R} = \{832 \times 480, 960 \times 544, 1056 \times 608, 1184 \times 672\}$, and $\gamma = 0.9$. To promote stable detector behavior across resolutions, we use multi-scale data augmentation during training (Sec. IV).

B. Framework Overview

Fig. 3 illustrates the overall architecture of SOAR. It has two modules, the MOT pipeline and the resolution adaptation module.

MOT Pipeline. The pipeline follows tracking-by-detection: a Resizer changes the input resolution according to action a_t, a Detector runs on the resized image, and an Associator links detections across frames. These components can use

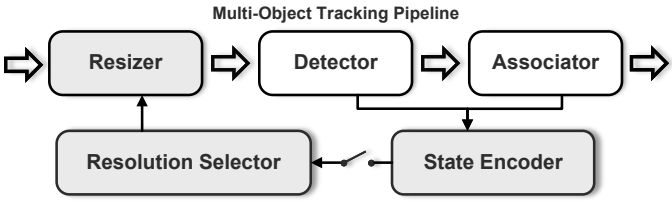

Fig. 3. Overall architecture of the SOAR framework.

standard detectors such as YOLO variants and association algorithms such as ByteTrack or BoTSORT; in our experiments, we instantiate the pipeline with YOLOv11s and BoTSORT (Sec. IV).

Resolution Adaptation Module. The State Encoder extracts a 9-dimensional state vector s_t from Detector and Associator outputs over the previous window, using proxy features correlated with object scale (e.g., detection confidence) to infer scene conditions without explicit distance estimation. The Resolution Selector then takes s_t and outputs action a_{t+1} for the next window. To learn the policy, we employ Conservative Q-Learning (CQL) [15] for offline reinforcement learning, which penalizes out-of-distribution actions to reduce value overestimation. At deployment, inference requires only a single forward pass through a lightweight multi-layer perceptron (MLP) and adds negligible overhead. Since SOAR does not constrain the choice of detector or tracker, it can be integrated with common MOT pipelines.

C. State Space Design

The state representation follows two principles. First, **indirect perception** uses proxy features correlated with object scale and avoids explicit distance measurement. Second, **computational efficiency** computes features from existing detector and tracker outputs without additional networks.

We use a 9-dimensional state vector $s_t \in \mathbb{R}^9$.

$$s_t = [\, r(a_t), v_{\text{lap}}, m_{\text{fpn}}, \bar{c}, c_{\min}, \bar{n}, m_{\text{mot}}, \ell_{\text{lost}}, a_{t-1} \,] \quad (1)$$

Each feature is computed over the previous W-frame window and normalized to $[0, 1]$ using minimum and maximum statistics computed on the training split. Values are clipped to the training range at test time.

- $r(a_t)$ denotes the current resolution index normalized by $|\mathcal{A}| - 1$ to the range $[0, 1]$.
- v_{lap} is the mean Laplacian variance across frames and measures image sharpness.
- m_{fpn} is the mean activation magnitude of the P5 feature map, which is the stride 32 output in YOLO after activation and averaged over channels and spatial locations.
- $\bar{c}$ and $c_{\min}$ are the mean and minimum detection confidence over all detected boxes.
- $\bar{n}$ is the mean object count per-frame in the window.
- m_{mot} is the mean bounding box motion magnitude computed as pixel displacement between matched boxes given by the tracker association in consecutive frames.
- ℓ_{lost} is the fraction of tracks marked as "lost" that are unmatched for more than 3 frames at the window end.

- a_{t-1} is the previous action normalized by $|\mathcal{A}|-1$ to $[0,1]$.

Among these features, **detection confidence** is an informative proxy for scene difficulty. Low confidence often indicates insufficient object scale or adverse imaging conditions. Including the **previous action** encourages temporal smoothness and reduces oscillation near decision boundaries.

D. Offline Dataset Construction

Reference Action Generation. Offline reinforcement learning depends on data coverage, so we construct a dataset $\mathcal{D}$ covering diverse UAV tracking conditions. For each temporal window in the training set, we run the full tracking pipeline at all candidate resolutions and compute window-level accuracy by accumulating true positives, false positives, false negatives, and identity switches within the W-frame window. We then select the lowest-cost resolution whose window-level MOTA remains within 1.0 percentage points of the highest-resolution setting. This reference action serves as a guidance signal for policy learning. Note that ground-truth annotations and any lookahead terms in the reward are used only during offline dataset construction; at inference, the policy observes only past-window statistics.

Trajectory Diversification. To improve state coverage, we uniformly sample four heuristic augmentation strategies: aggressive exploration that probes compute-saving limits, conservative verification that exposes wasted compute, temporal patterns that include planned resolution changes, and negative sampling that discourages unstable switching.

E. Reward Function Design

The reward jointly optimizes tracking accuracy and computational efficiency: $R = R_{\text{align}} + R_{\text{grad}} + R_{\text{corr}} - C_{\text{eff}}$. Let a_t^* denote the reference action for window t (Sec. III-D), and define the deviation $\delta_t = a_t - a_t^*$. The scoring function $S(\delta) = \max(0, 1 - \lambda|\delta|)$ assigns maximal score to exact matches and decays linearly with deviation magnitude.

Accuracy Terms. R_{align} is a lookahead-based potential function (used only during offline training) that rewards actions aligned with near-future scene changes:

$$R_{\text{align}} = \alpha_1 S(\delta_{t+1}) + \alpha_2 S(\delta_t) + \alpha_3 \sum_{k=1}^{3} \gamma_k S(a_{t+1} - a_{t+k}^*) \quad (2)$$

where α_1, α_2, and α_3 weight the one-step-ahead alignment, current-step alignment, and multi-step temporal alignment terms, respectively, and γ_k are temporal decay factors. To provide a dense optimization signal, $R_{\text{grad}} = S(\delta_{t+1}) - S(\delta_t)$ rewards error reduction and penalizes error increase.

Reference Guidance. R_{corr} leverages expert knowledge from the offline dataset. We set $R_{\text{corr}} = -\mathbb{I}(|\delta_t| \geq 2)$, which penalizes clearly incorrect decisions (deviation ≥ 2 resolution levels) and provides an implicit bonus when the policy corrects from erroneous states.

Efficiency Constraint. C_{eff} jointly constrains resolution selection and switching frequency:

$$C_{\text{eff}} = w_{\text{comp}} \cdot \text{cost}(r_t) + \beta \cdot \mathbb{I}(a_t \neq a_{t-1}) \quad (3)$$

where $\text{cost}(r_t) = \frac{w_t \cdot h_t}{1184 \times 672}$ is the normalized compute cost (input area divided by the baseline resolution area), and β controls the switch penalty strength.

Operating Modes. By adjusting w_{comp}, SOAR supports two modes: Conservative mode prioritizes accuracy with modest efficiency gains, and Aggressive mode pursues larger compute savings. Specific parameter values are provided in Sec. IV.

IV. Evaluation

A. Datasets

We evaluate our method on two representative benchmarks: VisDrone2019-MOT [5] and UAVDT [6]. VisDrone2019-MOT is a comprehensive dataset containing 56 training and 17 testing sequences. Following the official protocol, we use all 10 classes for training and report performance on the 5 core classes (car, bus, truck, pedestrian, and van). UAVDT focuses on vehicle tracking scenarios spanning a range of flight altitudes. We evaluate on its three target categories (car, truck, and bus). To ensure consistent evaluation, we compute results for both datasets using the official VisDrone2019-MOT toolkit, ensuring that the same metric implementation is applied throughout.

B. Metrics and Computational Baseline

We report MOTA and IDF1 to evaluate tracking accuracy and identity preservation, respectively, and use GFLOPs to quantify efficiency. We use the FLOPs of YOLOv11s [16] at 640×640 as a reference and estimate GFLOPs for other resolutions based on input size, following the common approximation that FLOPs scale approximately with image area. We report detector compute only, since (i) the overhead from BoTSORT association and the lightweight policy network is negligible relative to the detector, and (ii) frame resizing is a standard preprocessing step for UAV datasets with heterogeneous native resolutions. Therefore, preprocessing is treated consistently across fixed- and dynamic-resolution settings and is omitted from our GFLOPs accounting.

C. Implementation Details

Policy Training. The policy is learned via Offline Reinforcement Learning using the Discrete Conservative Q-Learning (CQL) algorithm. It employs a double Q-network structure, where each network is a three-layer MLP mapping the 9-dimensional state to a 256-dimensional hidden representation. The model is trained for 100,000 iterations using an Adam optimizer with a learning rate of 1×10^{-4} and a regularization coefficient of 0.5, on the pre-constructed offline transition dataset described in Sec. III-D.

Detector and Tracker. We employ the YOLO11 detector (initialized with the 's' variant of official pre-trained weights) and fine-tune it using an equal-probability random resize augmentation strategy. During deployment, the decision interval is set to 20 frames to smooth observation noise. The confidence and IoU thresholds are set to 0.2 and 0.45, respectively. Detection boxes are mapped back to the original resolution coordinates before being fed into the BoTSORT

tracker (ReID-free implementation from Ultralytics) to ensure spatial consistency.

Reward Configuration. The scoring function uses decay rate λ=0.5. The alignment reward employs coefficients α_1=0.75, α_2=−0.10, α_3=0.40 with temporal decay factors $\gamma_k \in \{0.5, 0.3, 0.2\}$ for $k \in \{1, 2, 3\}$. The efficiency term C_{eff} combines resolution cost and switch penalty (β=0.1). For Conservative mode, w_{comp}=0.3; for Aggressive mode, w_{comp}=0.6.

System Overhead. The inference overhead of the policy network is negligible (∼0.14 MFLOPs), constituting approximately 0.00035% of the YOLOv11s detector's computational load, and is therefore omitted from our GFLOPs accounting.

TABLE I
PERFORMANCE AND EFFICIENCY ON VISDRONE2019-MOT AND UAVDT. PARENTHESES DENOTE POINT DROPS RELATIVE TO THE 1184 × 672 BASELINE.

Dataset	Strategy	MOTA (%)	IDF1 (%)	Savings (%)
VisDrone	Baseline (1184 × 672)	48.30	56.8	-
	SOAR-Conservative	47.87 (↓ 0.43)	56.6 (↓ 0.20)	11.44%
	SOAR-Aggressive	47.69 (↓ 0.61)	56.8 (↓ 0.00)	20.36%
UAVDT	Baseline (1184 × 672)	54.67	71.4	-
	SOAR-Conservative	53.49 (↓ 1.18)	70.6 (↓ 0.80)	21.28%
	SOAR-Aggressive	52.15 (↓ 2.52)	70.0 (↓ 1.40)	44.42%

D. Dynamic Resolution Performance

Quantitative Results. Results in Table I suggest that not every frame requires the highest input fidelity. In VisDrone2019-MOT, where small objects are dense, SOAR-Conservative achieves 11.44% compute savings with only a 0.43-point MOTA drop and a 0.20-point IDF1 drop, indicating that the accuracy–efficiency trade-off is tighter on VisDrone2019-MOT. Notably, SOAR-Aggressive attains 20.36% savings while preserving IDF1 (0.00-point change) with a modest 0.61-point MOTA drop, suggesting that identity consistency is largely preserved in this setting even under more aggressive downscaling. In UAVDT, where vehicles are more prominent and backgrounds are simpler, the learned policy selects low resolutions more frequently, which yields larger compute savings (44.42%) with a moderate accuracy drop. This dataset-dependent behavior indicates that SOAR learns a context-conditioned allocation of compute. This adaptive capability is visually detailed in Fig. 5 ((e.g., uav0000201_00000_v), where the policy prefers higher resolutions under dynamic scale changes but converges to low resolutions under stable conditions.

Policy Behavior (Qualitative). To interpret the decision behavior, we visualize sample frames in Fig. 4 and analyze the action distributions in Fig. 5. In uav0000201_00000_v (Fig. 4a), the drone undergoes dynamic altitude and viewpoint changes, causing object scale and scene difficulty to vary over time. Accordingly, the agent adopts a mixed strategy with a clear preference for high resolution (Res 1184 × 672 selected ≈44%) to maintain tracking stability, while still exploiting lower resolutions when the state indicates stable tracking. In contrast, in uav0000077_00720_v (Fig. 4b), the visual context represents a stable low-altitude cruising scenario with large

Fig. 4. Example frames (50, 350, and 650) from two sequences that illustrate temporal differences in viewpoint and scene complexity (left: uav0000201_00000_v; right: uav0000077_00720_v).

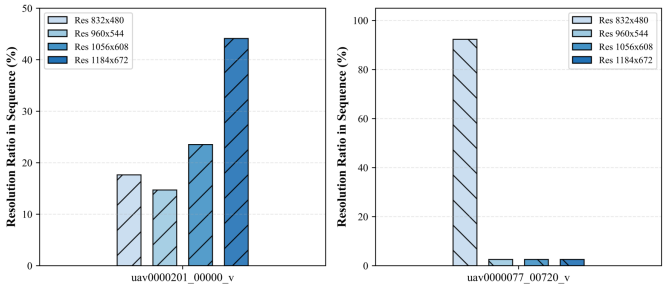

Fig. 5. Resolution selection frequency on VisDrone for the two sequences in Fig. 4: (left) uav0000201_00000_v, (right) uav0000077_00720_v.

targets. Given this consistent redundancy, the policy quickly converges to the minimum resolution, selecting Res 832 × 480 in > 90% of frames. Importantly, these two cases demonstrate that SOAR does not simply default to a fixed setting: it raises resolution in response to scale or viewpoint shifts (to preserve detection and association reliability) and stabilizes at low resolution only when the visual context remains stable over time.

E. Sensitivity Analysis

We analyze resolution sensitivity in Table II. The non-linear trend yields three observations:

(i) **Breakdown threshold.** Performance drops sharply when reducing resolution below 832 × 480 (VisDrone: ΔMOTA = −4.20 and ΔIDF1 = −2.38 from 832 × 480 to 736 × 416). This suggests that, at low resolutions, the pixel support for small/distant targets becomes insufficient for reliable detection and association, leading to markedly more tracking errors.

(ii) **Saturation.** Increasing resolution beyond 1184 × 672 substantially increases compute (e.g., +50.65% GFLOPs) but yields only marginal accuracy gains (e.g., ≤ 0.6 MOTA points on UAVDT), suggesting that performance is increasingly limited by factors such as occlusion and motion blur rather than input pixel density.

(iii) **Dynamic scheduling vs. static scaling.** At similar compute budgets, context-aware scheduling yields a better accuracy–efficiency trade-off than a single static resolution. For example, compared with the closest fixed setting (1056 × 608, 19.32% savings), SOAR-Aggressive achieves comparable savings (20.36%) while improving VisDrone performance by 0.73 MOTA and 0.87 IDF1.

TABLE II
RESOLUTION SENSITIVITY ANALYSIS ON TRACKING PERFORMANCE.

Resolution ($W \times H$) lr)4-5 lr)6-7	GFLOPs	Savings (%)	UAVDT		VisDrone	
			MOTA	IDF1	MOTA	IDF1
1440 × 832	57.7 G	−50.65	55.21	71.77	48.48	57.09
1344 × 768	49.7 G	−29.77	55.02	71.94	48.34	57.14
1280 × 736	45.4 G	−18.54	55.10	71.97	48.25	56.89
1184 × 672	38.3 G	0.00 (Base)	54.67	71.43	48.30	56.83
1056 × 608	30.9 G	19.32	53.80	71.13	46.96	55.93
960 × 544	25.2 G	34.20	53.10	70.29	45.57	54.81
832 × 480	19.2 G	49.87	51.30	69.63	44.26	53.30
736 × 416	14.8 G	61.36	50.25	68.85	40.06	50.92
640 × 384	11.8 G	69.19	48.98	68.28	40.00	50.92

TABLE III
ABLATION STUDY OF DIFFERENT REWARD COMPONENTS ON THE VISDRONE2019-MOT DATASET.

Configuration	MOTA (%)	IDF1 (%)	Savings (%)
Full SOAR	47.87	56.60	11.44
without R_{align}	46.90	56.03	15.82
without R_{grad}	47.07	56.06	14.72
without R_{corr}	47.94	56.43	1.64
without C_{eff}	48.30	56.83	0.00

F. Ablation Study

Table III reveals complementary roles of the reward components. Removing the efficiency constraint C_{eff} eliminates savings (0.00%), confirming that it is the primary driver for reducing compute; correspondingly, MOTA/IDF1 peak because the policy collapses to the highest-resolution default. In contrast, removing R_{corr} collapses savings (11.44% $\rightarrow$ 1.64%) while keeping both MOTA and IDF1 nearly unchanged (MOTA: 47.87 $\rightarrow$ 47.94; IDF1: 56.60 $\rightarrow$ 56.43), which is consistent with a conservative fallback behavior that defaults to higher resolutions when the reward no longer provides an explicit signal about safe downscaling opportunities. Finally, R_{align} and R_{grad} act as quality regularizers: removing either increases savings but reduces both MOTA and IDF1, indicating that overly aggressive downscaling degrades detection/association quality. Overall, the full reward achieves the best accuracy–efficiency balance.

V. CONCLUSION

We present SOAR (Scene-Aware Adaptive Resolution), a framework for UAV multi-object tracking that treats input resolution as a sequential decision variable optimized via offline reinforcement learning. By inferring scene conditions from proxy signals such as detection confidence, SOAR exploits resolution redundancy without explicit distance estimation. On VisDrone2019-MOT and UAVDT, the conservative setting achieves 11%–21% GFLOPs reduction with MOTA drops under 1.2 points; the aggressive setting reaches 20%–44% reduction with MOTA drops under 2.5 points, while IDF1 remains nearly unchanged. Ablation confirms that efficiency constraints drive compute savings and accuracy terms prevent excessive downscaling. The learned policy adapts resolution to scene dynamics rather than relying on a fixed setting, and integrates readily into standard tracking-by-detection pipelines.

ACKNOWLEDGMENT

This work is supported by the Guangxi Key Research and Development Program (No. AB24010300), the National Natural Science Foundation of China (Grant Nos. 62362011 and 62462016), the Guangxi Postgraduate Education Reform Project (No. JGY2022124), and the Project of the Guangxi Key Laboratory of Image and Graphic Intelligent Processing (No. GIIP2206).

This paper was prepared with the assistance of AI-based language tools for editing and polishing. All technical content, experimental design, and scientific conclusions are the sole responsibility of the authors.

REFERENCES

[1] Song Han, Huizi Mao, and William J Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *International Conference on Learning Representations*, 2016.

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[3] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang, "Resolution adaptive networks for efficient inference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2369–2378.

[4] Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang, "Glance and focus: a dynamic approach to reducing spatial redundancy in image classification," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 2432–2444.

[5] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7380–7399, 2021.

[6] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian, "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 370–386.

[7] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *European Conference on Computer Vision*. Springer, 2022, pp. 1–21.

[8] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," in *arXiv preprint arXiv:2206.14651*, 2022.

[9] Peng Wang, Yongcai Wang, and Deying Li, "DroneMOT: Drone-based multi-object tracking considering detection difficulties and simultaneous moving of drones and objects," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 7397–7404.

[10] Yujin Zheng, Chu He, Xiaohan Chen, Huan Zhang, Tao Qu, and Dingwen Wang, "DFA-MOT: A dynamic field-aware multi-object tracking framework for uncrewed aerial vehicles," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 11, pp. 11350–11361, 2025.

[11] Xiang Li, Cheng Chen, Yuan-yao Lou, Mustafa Abdallah, Kwang Taik Kim, and Saurabh Bagchi, "Hoptrack: A real-time multi-object tracking system for embedded devices," *arXiv preprint arXiv:2411.00608*, 2024.

[12] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica, "Chameleon: scalable adaptation of video analytics," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 253–266.

[13] Keivan Nalaie, Renjie Xu, and Rong Zheng, "Deepscale: Online frame size adaptation for multi-object tracking on smart cameras at the edge," *ACM Transactions on Internet of Things*, vol. 3, no. 4, pp. 1–27, 2022.

[14] Renjie Xu, Keivan Nalaie, and Rong Zheng, "Fasttuner: Fast resolution and model tuning for multi-object tracking in edge video analytics," *IEEE Transactions on Mobile Computing*, 2025.

[15] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine, "Conservative q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1179–1191.

[16] Glenn Jocher, Ayush Chaurasia, and Jing Qiu, "Ultralytics YOLO11," https://github.com/ultralytics/ultralytics, 2024, Accessed: 2025-01-15.