CGTrack: Cascade Gating Network with Hierarchical Feature Aggregation for UAV Tracking

Weihong Li[1], Xiaoqiong Liu[2], Heng Fan[2,†], and Libo Zhang[3,†,*]

Abstract— **Recent advancements in visual object tracking have markedly improved the capabilities of unmanned aerial vehicle (UAV) tracking, which is a critical component in real-world robotics applications. While the integration of hierarchical lightweight networks has become a prevalent strategy for enhancing efficiency in UAV tracking, it often results in a significant drop in network capacity, which further exacerbates challenges in UAV scenarios, such as frequent occlusions and extreme changes in viewing angles. To address these issues, we introduce a novel family of UAV trackers, termed CGTrack, which combines explicit and implicit techniques to expand network capacity within a coarse-to-fine framework. Specifically, we first introduce a Hierarchical Feature Cascade (HFC) module that leverages the spirit of feature reuse to increase network capacity by integrating the deep semantic cues with the rich spatial information, incurring minimal computational costs while enhancing feature representation. Based on this, we design a novel Lightweight Gated Center Head (LGCH) that utilizes gating mechanisms to decouple target-oriented coordinates from previously expanded features, which contain dense local discriminative information. Extensive experiments on three challenging UAV tracking benchmarks demonstrate that CGTrack achieves state-of-the-art performance while running fast. Code will be available at https://github.com/Nightwatch-Fox11/CGTrack.**

I. Introduction

Unmanned aerial vehicles (UAVs) commonly refer to drones remotely operated by a human operator without any pilot on board. The rapid development of UAVs has boosted numerous real-world applications, *e.g.*, logistic and product deliveries [1], UAV-assisted IoT applications [2], and robotic automation [3]. Despite the remarkable progress in visual object tracking, achieving efficient and accurate UAV tracking remains fraught with significant challenges, such as frequent scale changes, extreme viewing angles, and severe occlusions. These issues are particularly pronounced in the context of fast-moving drones. Therefore, it is crucial to develop more robust and efficient network designs, especially for edge devices with limited power resources.

In general, the majority of UAV trackers can be categorized into two types: discriminative correlation filters (DCF)-based trackers [4]–[9] or deep learning (DL)-based trackers [10]–[16]. Despite the superior efficiency brought by Fourier transformation, the accuracy of DCF-based trackers has fallen far behind that of the DL-based trackers.

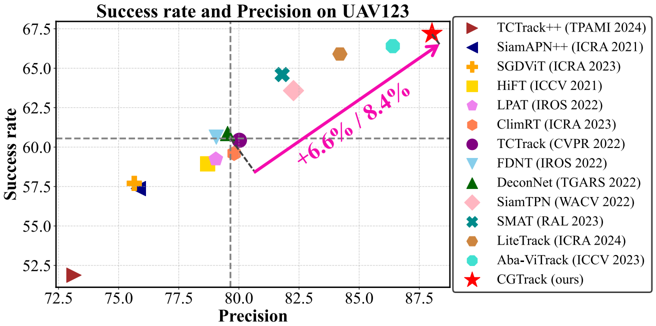

Fig. 1. Comparison of success rate and precision between CGTrack and other 13 state-of-the-art (SOTA) trackers on the authoritative UAV123 benchmark [34]. CGTrack achieves SOTA performance in both precision and success rate, surpassing the average performance of 13 trackers by **6.6%** and **8.4%** respectively. Best viewed in color for all figures in this paper.

Among modern DL-based trackers, those based on Siamese networks [10]–[12], [17]–[20] are the most prevalent. They employ the strategy of "divide-and-conquer", where the template and the search region features are extracted separately before relation modeling. However, as described in [15], the features extracted by Siamese networks lack essential target-oriented discriminative information, resulting in significant performance degradation in UAV scenarios, particularly during high-speed movement. Recently, Transformer [21] has played a pivotal role in the field of UAV tracking [22]–[24] due to its superior capacity of modeling global relationships. Moreover, with the power of pre-trained Vision Transformer (ViT) models [25]–[29], one-stream frameworks [15], [30]–[32] exhibit superior performance in both accuracy and efficiency when compared to Siamese-based trackers. However, Transformer-based trackers are burdened by high computational costs brought by ViTs and often neglect critical local information [33], leading to failures in extreme UAV scenarios.

In this work, we address the aforementioned challenges by introducing a lightweight, hierarchical one-stream tracking framework. By adopting a lightweight hierarchical ViT as backbone, we obtain hierarchical features that preserve rich global contextual information. To effectively enhance network capacity with these hierarchical features, we propose a Hierarchical Feature Cascade (HFC) module, inspired by DenseNet [35], which highlights the strength of feature reuse. However, unlike the original DenseNet [35], our HFC module simplifies dense connections into a cascade structure by scaling multi-level features to a uniform size and later concatenating them. This approach allows us to obtain a feature map containing both deep semantic information and

[†]Equal advising and co-last authors; [*]Corresponding author

[1]Weihong Li is with the Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou 310024, China.

[2]Xiaoqiong Liu and Heng Fan are with the Dept. of Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA.

[3]Libo Zhang is with the Institute of Software Chinese Academy of Science, Beijing 100190, China. libo@iscas.ac.cn

shallow detail information without additional parameters or FLOPs. The HFC module explicitly increases the network width (*i.e.*, channel number) to provide rich contextual information for subsequent fine-grained discriminative feature extraction. The enriched feature map contains both discriminative local details and global context which are particularly crucial in resource-constrained challenging UAV scenarios. Additionally, we introduce a Residual Squeeze-and-Excitation (SE) module to apply coarse-grained gating on the feature maps after each concatenation, improving gradient flow through its residual design.

To fully leverage the feature map generated by the HFC module, we improve the center head design in modern trackers [15], [36]–[38] by proposing the Lightweight Gated Center Head. Inspired by [39], we replace the basic Conv-BN-ReLU (CBR) block with an Efficient Gating (EG) block. The EG block first maps the features into a high-dimensional nonlinear feature space, where gating is performed subsequently via the Hadamard product. The gating mechanisms have shown superior capability in enhancing local fine-grained details in the field of image inpainting [40]. In light of this, we employ EG block to further mine local discriminative information which is critical for addressing the challenges inherent in UAV scenarios.

In summary, our contributions in this paper are as follows:

- We propose CGTrack, a family of UAV tracking architecture aiming at combining global context provided by lightweight ViT with mined local discriminative details from hierarchical features to achieve robust UAV tracking.
- Leveraging the art of feature reuse, a novel HFC module is presented whereby hierarchical features are aggregated and gated in an efficient cascade pipeline.
- An original tracking head LGCH is introduced. It further utilizes the HFC-expanded features by mapping the features to a higher-dimensional nonlinear feature space, whereby gating is performed through the Hadamard product.
- We perform comprehensive evaluations on three authoritative UAV tracking benchmarks demonstrating the state-of-the-art performance of CGTrack.

II. Related Works

A. Visual Object Tracking for UAV.

Despite the high efficiency of DCF-based trackers [4]–[9], they have been largely supplanted by Siamese-based trackers [10]–[16] in UAV tracking due to their relatively low accuracy. More recently, some studies have attempted to incorporate Transformer [21] into Siamese-based UAV tracking pipeline [22]–[24] to enhance the interaction between extracted template and search region features. However, these methods still adopt the two-stream framework, leading to insufficient information interaction during feature extraction. In this work, we adapt a lightweight hierarchical ViT into the one-stream UAV tracking pipeline, establishing an optimal equilibrium between computational demands and tracking accuracy

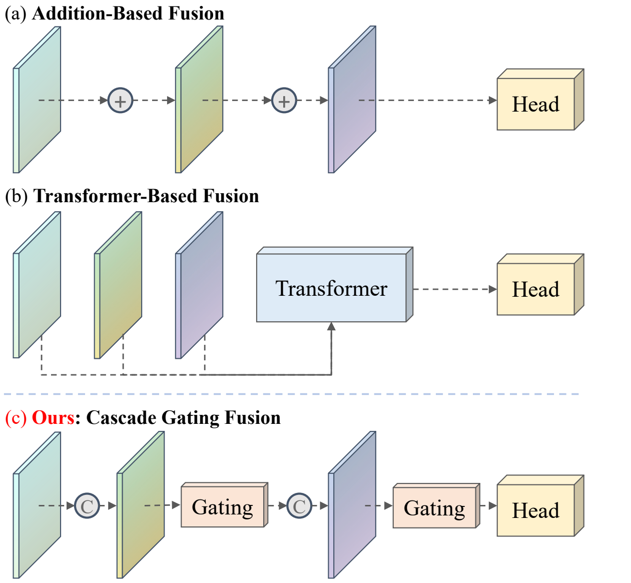
(a) Addition-Based Fusion
Head
(b) Transformer-Based Fusion
Transformer
Head
(c) Ours: Cascade Gating Fusion
C
Gating
C
Gating
Head

Fig. 2. Comparison of the popular hierarchical feature fusion methods for UAV tracking. **(a)** Addition-Based Fusion: simply adds all the feature maps up. **(b)** Transformer-Based Fusion: employs Transformer layers or multi-head attention modules for feature fusion. **(c)** Cascade Gating Fusion: concatenates adjacent feature maps and performs gating subsequently in a cascade architecture.

B. Hierarchical UAV Trackers.

In UAV scenarios, most existing trackers adopt lightweight image classification networks as backbones for better computational efficiency [10], [22]–[24], [41]–[43]. However, the high-stride downsampling in these networks often leads to a loss of critical information. To mitigate this, hierarchical UAV trackers attempt to leverage multi-level feature maps generated at different stages of the lightweight backbones. For instance, SiamAPN++ [41] employs an attention mechanism to adaptively fuse multi-level features, while HiFT [22] stacks Transformer layers to incorporate multi-scales feature maps. These methods, as shown in Fig. 2(b), are burdened with heavy relation modeling. Beyond UAV-specific trackers, HiT [32] employs simple addition for hierarchical feature aggregation, also depicted in Fig. 2(a). This approach, despite its simplicity, ignores the diverse variance of hierarchical feature maps, resulting in severe information loss. As shown in Fig. 2(c), unlike the aforementioned methods, we propose a novel cascade gating structure that combines gating mechanism and feature reuse to expand network capacity with minor computational costs.

C. Gating Mechenism.

Recent studies have demonstrated the utility of the gating mechanism across numerous computer vision tasks [39], [40], [44], [45]. To illustrate, SENet [44] introduces an efficient channel-wise attention mechanism through lightweight gating. DeepFill v2 [40] incorporates the gating mechanism into the convolution to better distinguish between different pixels in an image. Moreover, StarNet [39] further explains the reason why numerous efficient network designs adopt gating mechanisms [44], [45]: the Hadamard product has

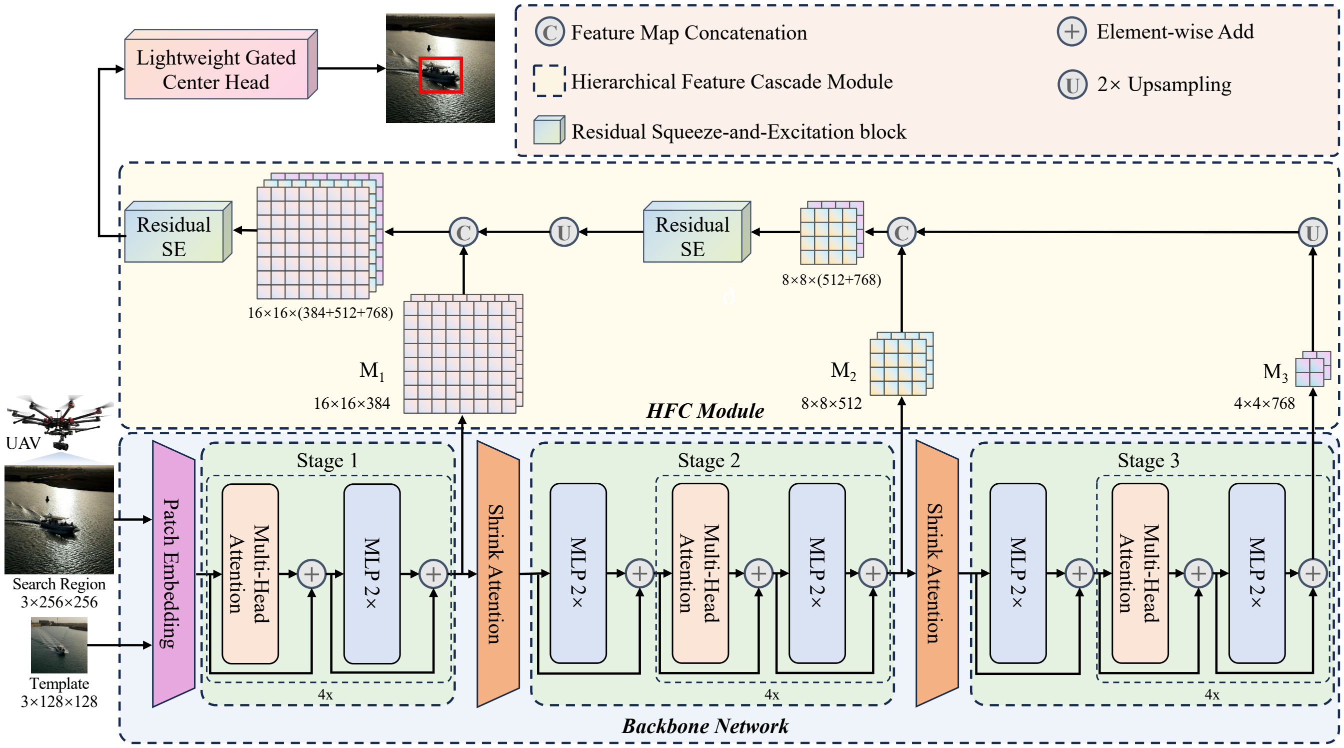

Fig. 3. Overview of the proposed CGTrack, which comprises three main components: a lightweight hierarchical backbone, an HFC module, and a Lightweight Gated Center Head.

the ability to map the features into higher and nonlinear dimensions, but operates in low-dimensional space. In this work, we integrate the coarse-gained gating and fine-grained gating into the tracking framework respectively to mine local discriminative details with efficiency.

III. Proposed Method

This section systematically elaborates on our proposed CGTrack framework. We first establish a conceptual schematic of the architecture, followed by the detailed component introduction, including a lightweight ViT backbone, the proposed *Hierarchical Feature Cascade* module, and *Lightweight Gated Center Head*. In the final part of this section, we introduce the training objective.

A. Overview

As depicted in Fig. 3, CGTrack is a one-stream tracking framework incorporating three core components: a lightweight ViT backbone, the proposed HFC module, and LGCH. Like most one-stream trackers [15], [16], [32], [46], CGTrack takes a pair of images as input and then jointly performs feature extraction and relational modeling across different network stages. The model progressively generates hierarchical feature maps with varying spatial resolutions from each ViT stage. Besides the backbone is a coarse-to-fine architecture consisting of our proposed HFC module and LGCH: The feature sequence is firstly fed into the HFC module for efficient feature augmentation and preliminary gating; Then, LGCH takes the expanded feature map as input and performs final purification to obtain tracking result.

B. LeViT Backbone

Inspired by HiT [32], we adopt LeViT [47] as the backbone of CGTrack and adapt it into the tracking framework. For clarification, we denote the input template image and search region image as $\mathbf{Z} \in \mathbb{R}^{3\times H_z\times W_z}$ and $\mathbf{X} \in \mathbb{R}^{3\times H_x\times W_x}$ respectively. They are first downsampled by a factor of 16 through patch embedding resulting in $\mathbf{Z_p} \in \mathbb{R}^{C\times\frac{H_z}{16}\times\frac{W_z}{16}}$ and $\mathbf{X_p} \in \mathbb{R}^{C\times\frac{H_x}{16}\times\frac{W_x}{16}}$. Then we flatten and concatenate $\mathbf{Z_p}$ and $\mathbf{X_p}$ in the spatial dimension and feed them into the following ViT stages. The transformer part of LeViT comprises three stages, and each stage consists of Li blocks, *i.e.*, $L1$=4, $L2$=4, $L3$=4. Each block has a Multi-Head Attention and an MLP in the residual form. LeViT leverages the Shrink Attention modules to downsample feature maps at a scale of 4 between stages, producing three feature maps with multiple resolutions. As is common in one-stream trackers, we extract the search region part of the output from each stage and re-interpret these tokens to a 2D spatial correlation map. Finally, we obtain a sequence including three correlation maps with distinct size, *i.e.*, $\mathbf{M_1} \in \mathbb{R}^{H_s\times W_s\times C_s}$, $\mathbf{M_2} \in \mathbb{R}^{H_m\times W_m\times C_m}$, $\mathbf{M_3} \in \mathbb{R}^{H_l\times W_l\times C_l}$, where $C_s = 384$, $C_m = 512$, $C_l = 768$. In addition, a similar position encoding design, analogous to Dual-image Position Encoding in HiT is used in our CGTrack, to better adapt LeViT for the tracking task. Further details regarding the backbone network of CGTrack can be found in LeViT and HiT.

Remark 1: Attributing to the lightweight hierarchical ViT, we obtain a sequence of correlation maps preserving rich global context information in both search region and template with minor costs. This highly parallelized one-stream structure is able to handle multiple UAV scenarios with flexible variants.

C. Hierarchical Feature Cascade Module

Inspired by DenseNet [35], we propose a Hierarchical Feature Cascade module that progressively integrates multi-stage backbone features via concatenation operations. By applying gating to the concatenated feature maps, the HFC module enhances the critical local discriminative details. Different from the original dense connection in DenseNet, the HFC module simplifies it by only keeping one cascade path between adjacent feature maps, which significantly reduces memory usage and achieves promising efficiency. As illustrated in Fig. 3, for hierarchical 3D correlation maps denoted as $M_i,\ i \in \{1, 2, 3\}$, we first upsample M_1 and concatenate it and M_2 together along the channel dimension, which can be written as

$$X = \text{Concat}(\mathbf{M_2}, \text{Upsample}(\mathbf{M_1})) \tag{1}$$

where X is the intermediate result of the HFC module. In view of the diverse variance of the concatenated features, we adapt the original Squeeze-and-Excitation block [44] into the residual form and propose Residual SE, which efficiently applies channel re-scaling and preliminary gating to the concatenated feature map through Hadamard product. The entire process of Residual SE can be formulated as

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{c,i,j} \tag{2}$$

$$s_c = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot z_c)), \tag{3}$$

$$\hat{x}_{c,i,j} = x_{c,i,j} \cdot s_c, \tag{4}$$

$$\hat{X} = X + X \odot S \tag{5}$$

where $x_{c,i,j}$ represents the input feature at channel c and spatial location (i, j). z_c is the channel descriptor obtained via global average pooling. W_1 and W_2 are the weights of the fully connected layers used in the excitation step. σ is the sigmoid activation function. X and $\hat{X}$ are the overall input and output feature maps, respectively. $\odot$ denotes Hadamard product. Then, we apply the same operation to the output of the previous step to obtain the final feature. The entire process can be mathematically formulated as:

$$O = \text{ResidualSE}(\text{Concat}(\mathbf{M_2}, \text{Upsample}(\mathbf{M_1}))) \tag{6}$$

$$Y = \text{ResidualSE}(\text{Concat}(\mathbf{M_3}, \text{Upsample}(\mathbf{O}))) \tag{7}$$

Remark 2: Compared to additive operations, feature concatenations preserve all information and improve gradient flow, thereby accelerating network convergence. Each concatenation operation yields a higher-dimensional feature map explicitly. By applying the Residual SE to the concatenated feature map, we enhance local discriminative details through gating. This amplification is particularly crucial in challenging UAV scenarios, such as tracking small objects or handling sudden appearance changes.

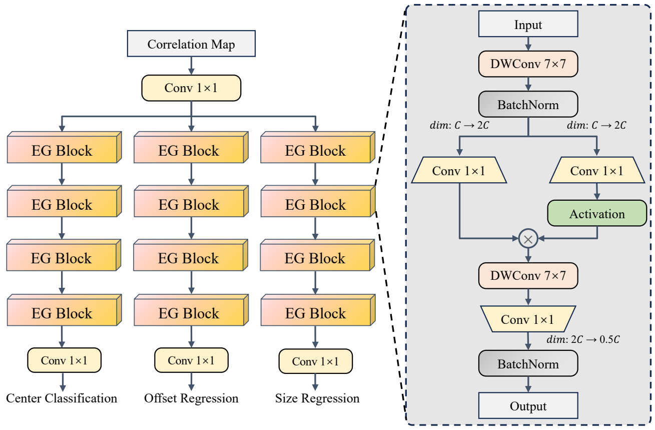

Fig. 4. Detailed architectures of LGCH. The left part illustrates the overall workflow of LGCH. The right one shows the structure of the EG block.

D. Lightweight Gated Center Head

From the HFC module, we obtain features enriched with contextual information. Building on this, we propose the Efficient Gating block to extract fine-grained discriminative details from the expanded features. As illustrated in Fig. 4, the concatenated feature map is first downsampled by a 1 × 1 convolution into a channel size of 256 for memory efficiency. The feature map is then fed into three branches, each containing four EG blocks followed by a 1 × 1 convolution. Consistent with [15], the output of three branches is a classification score map, a local offset map, and a bounding box size map, respectively. The detailed structure of the EG block is depicted on the right side of Fig. 4. The core design of the EG block involves two 1 × 1 convolutions that map the input into higher-dimensional, non-linear feature space. Among these, one branch incorporates an activation function, forming the *gate* branch while the other serves as the *context* branch. Subsequently, the Hadamard product is performed between *gate* and *context*. The entire process of the EG block can be written as:

$$O = \text{BN}(\text{DW}_{7\times7}(X)) \tag{8}$$

$$X_1 = \text{Conv}_{1\times1}(O) \tag{9}$$

$$X_2 = \text{Conv}_{1\times1}(O) \tag{10}$$

$$P = \text{ReLU6}(X_1) \odot X_2 \tag{11}$$

$$Y = \text{BN}(\text{Conv}_{1\times1}(\text{DW}_{7\times7}(P))) \tag{12}$$

where X_1 and X_2 are the *gate* and *context* branch, respectively.

Remark 3: Unlike commonly used CBR blocks, the proposed EG blocks exhibit enhanced capability in extracting fine-grained features. Similar to the HFC module, EG blocks first explicitly map features to higher dimensions before performing gating, effectively decoupling discriminative, target-oriented information in the correlation map. Furthermore, as described in [39], the Hadamard product implicitly transforms previously expanded features into exceptionally high and nonlinear dimensions while maintaining operations in a low-dimensional space. Notably, our proposed EG block

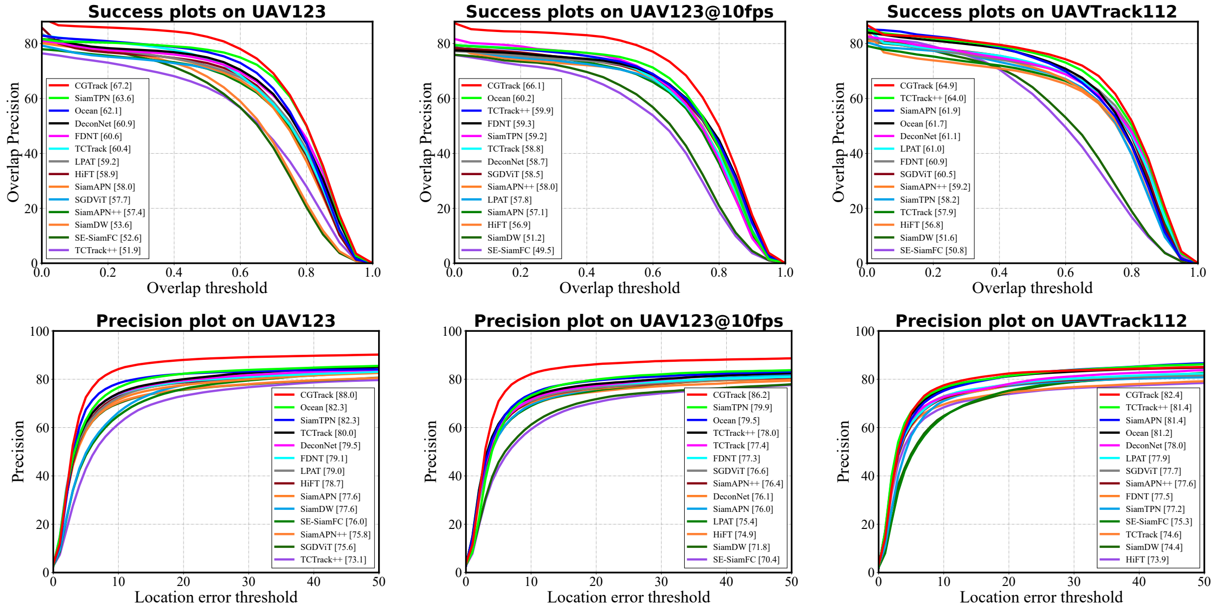

Fig. 5. Overall performance of CGTrack and prevailing SOTA trackers on UAV123 [34] (the first column), UAV123@10fp [34] (the second column), and UAVTrack112 [48] (the third column) benchmarks. CGTrack achieves SOTA performance across all benchmarks.

has even fewer FLOPs and parameters than a CBR block, highlighting its potential for applications on edge devices.

E. Training objective

In the training phase, we employ weighted focal loss [49] in the classification task, while utilizing a combination of ℓ_1 loss and generalized GIoU loss [50] for the localization task. The overall loss function is

$$\mathcal{L} = \mathcal{L}_{\text{focal}} + \lambda_G \mathcal{L}_{\text{GIoU}} + \lambda_l \mathcal{L}_l, \tag{13}$$

where $\lambda_G = 2$ and $\lambda_l = 5$ are the regularization parameters following [15].

IV. Experiments

A. Implementation Details

The proposed CGTrack is implemented in Python 3.8 with PyTorch 1.10.0, trained on four NVIDIA RTX 3090 GPUs. We employ the train-splits of GOT-10k [51] (excluding 1k sequences as convention), LaSOT [52], COCO2017 [53], and TrackingNet [54] for training. Common data augmentations including horizontal flipping and brightness jittering are applied during training. The network processes 128×128 template and 256×256 search images in each training batch of 128 samples. We adopt AdamW [55], with the weight decay of 1e-4 as the optimizer. The initial learning rate of CGTrack is set to 4e-5 which decays by 10% during the final 20% training epochs.

B. Overall Performance on UAV Tracking Benchmarks

In this subsection, our CGTrack is comprehensively compared with 13 SOTA trackers including TCTrack [42], SGDViT [23], FDNT [56], HiFT [22], SiamAPN [48], LPAT [57], DeconNet [58], SiamTPN [59], SiamDW [60], TCTrack++ [43], SiamAPN++ [41], SE-SiamFC [61], Ocean [62] on three public authoritative aerial tracking benchmarks.

UAV123. UAV123 [34] is a comprehensive aerial video benchmark dataset containing 123 HD sequences with over 112,000 frames captured from low-altitude aerial platforms. This benchmark includes various challenging scenarios such as rapid target motion and scale variation, providing a comprehensive platform to thoroughly evaluate CGTrack's performance in aerial tracking. As depicted in Fig. 5, CGTrack demonstrates SOTA tracking performance with 88.0% in Precision and 67.2% in Success score.

UAV123@10fps. UAV123@10fps [34] is derived by downsampling the original 30fps version, which leads to more pronounced motion between consecutive frames. This increased motion poses a challenge to trackers in more effectively leveraging inter-frame continuity information for robust aerial tracking. As shown in Fig. 5, CGTrack consistently achieves SOTA performance, with the highest Precision (83.8%) and Success score (66.1%).

UAVTrack112. UAVTrack112 [48] is a challenging aerial benchmark that collects 112 real-world sequences including low illumination scenarios in the dark time. As illustrated in Fig. 5, CGTrack sets a new SOTA Success score of 64.9% and Precision score of 80.6%.

C. Attribute-Based Comparison

To further evaluate the robustness of CGTrack against diverse UAV-specific challenges, we conduct exhaustive attribute-based comparisons with other 5 SOTA UAV trackers. As depicted in Fig. 6, our CGTrack achieves SOTA performance in all attributes. The promising results demonstrate that CGTrack is capable of aggregating the mined local discriminative details and global context to mitigate various challenges in UAV scenarios.

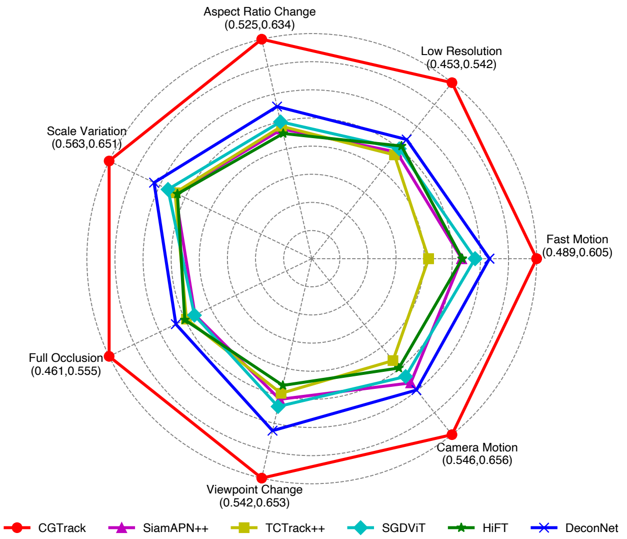

Fig. 6. Success scores of different attributes among top 6 SOTA UAV trackers. CGTrack significantly outperforms other trackers in typical UAV attributes.

#	Method	Precision	Params(M)	MACs(G)
1	Addition-based Fusion	82.90	40.931	4.315
2	Concatenation-based Fusion(w/o Residual SE)	84.12	40.668	4.323
3	Concatenation-based Fusion + Residual SE	**86.24**	41.219	4.324

TABLE I Ablation study of different fusion manners of CGTrack. Gray color is employed to denote our final configuration.

#	Method	Params(M)	MACs(G)	FPS	Precision
1	CGTrack-T	9.987	1.165	61.4	80.08
2	CGTrack-S	11.421	1.300	55.6	82.00
3	CGTrack-B	41.219	4.324	42.1	86.24

TABLE II Details and variants of our CGTrack model.

#	Method	Head Params(M)	MACs(G)	AUC
1	Plain CBR block	2.935	0.751	65.51
2	EG block-1x	1.425	0.364	63.64
3	EG block-2x	**2.665**	**0.680**	**66.14**
4	EG block-3x	3.904	0.996	65.31
4	EG block-4x	5.143	1.313	63.40

TABLE III Ablation study on different components and configurations of the center head. Gray denotes our final configuration.

D. Ablation Study and Visualization

In this subsection, we present the ablation studies on UAV123@10fps.

Hierarchical Feature Fusion Analysis. To verify the superiority of our fusion manner, we compare different hierarchical feature fusion manners. As shown in Tab. I, the original concatenation-based fusion (without gating) gains a 1.22% improvement in Precision score. Furthermore, as enumerated in Tab. I, Row 3, the cascade gating framework exhibits a 3.34% increase in Precision score compared to Row 2. The aforementioned results demonstrate the efficacy of the designed HFC module in UAV tracking.

Variants Analysis. In Tab. II, we present multiple variants of CGTrack with different backbone networks. Specifically, we adopt LeViT-384 [47], LeViT-128, and LeViT-128S for CGTrack-B, CGTrack-S, and CGTrack-T, respectively. Among these variants, CGTrack-T exhibits superior speed at 61.4 fps on an NVIDIA RTX 3090 GPU, while CGTrack-B focuses on robustness, achieving an 86.24% Precision score on UAV123@10fps. Notably, the CGTrack-S achieves a more favorable balance between computational complexity and tracking performance. The designed variants show strong generalization across different application scenarios.

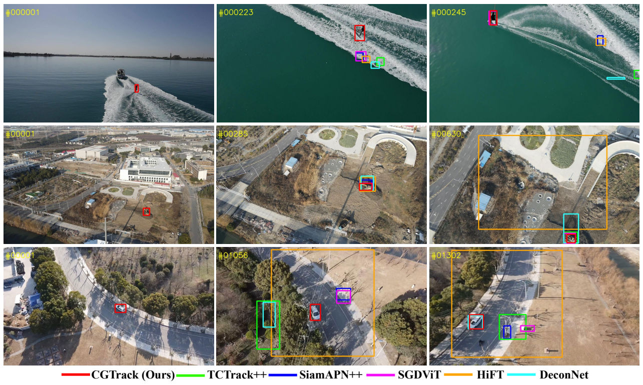

Fig. 7. Qualitative comparison of CGTrack with other trackers on three representative sequences (*wakeboard2* from UAV123@10fps, and *excavator*, *car4* from UAVTrack112). CGTrack achieves robust performance under severe UAV-specific challenges.

LGCH Analysis. This part analyzes the effectiveness of LGCH and compares different feature upsampling ratios in the EG block. As in Tab. III, LGCH outperforms the CBR-based center head with even fewer parameters and FLOPs. When setting the upsampling ratio to 2, the highest Success score is achieved. The results indicate that a larger upsampling ratio can cause overfitting, while a smaller upsampling ratio may result in insufficient modeling capacity.

Qualitative Results. To intuitively demonstrate the tracking performance in real-world scenarios, we visualize the tracking results in Fig. 7. The qualitative results across multiple real-world challenging scenes demonstrate that our CGTrack achieves superior robustness and accuracy, outperforming all the other UAV trackers. For further visualization of our method and comparison to other trackers, please kindly refer to the accompanying video.

V. Conclusion

In this work, we introduce a novel family of lightweight one-stream UAV trackers, dubbed CGTrack. CGTrack integrates global contextual information with mined local discriminative details to bridge the gap between lightweight ViTs and robust UAV tracking. By leveraging the art of feature reuse and gating mechanism, CGTrack significantly expands network capacity to tackle challenges in UAV scenarios without additional computational overhead. Extensive experiments demonstrate the superior real-world practicability and state-of-the-art performance of CGTrack. Finally, we hope this work could inspire and facilitate future research in robust UAV tracking.

ACKNOWLEDGMENT

Libo Zhang is supported by National Natural Science Foundation of China (No. 62476266). Heng Fan is not supported by any fund for this work.

References

[1] S. Kim and I. Moon, "Ieee tsmc," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 42–52, 2019.

[2] Q. Wang, H. N. Dai, X. Li, M. K. Shukla, and M. Imran, "Artificial noise aided scheme to secure uav-assisted internet of things with wireless power transfer," *Computer Communications*, vol. 164, 2020.

[3] Z. Ouyang, R. Mei, Z. Liu, M. Wei, Z. Zhou, and H. Cheng, "Control of an aerial manipulator using a quadrotor with a replaceable robotic arm," in *ICRA*. IEEE, 2021, pp. 153–159.

[4] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE TPAMI*, vol. 37, no. 3, pp. 583–596, 2014.

[5] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient Convolution Operators for Tracking," in *CVPR*, 2017, pp. 6931–6939.

[6] ——, "Atom: Accurate tracking by overlap maximization," in *CVPR*, 2019, pp. 4660–4669.

[7] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning Discriminative Model Prediction for Tracking," in *ICCV*, 2019, pp. 6181–6190.

[8] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic Regression for Visual Tracking," in *CVPR*, 2020, pp. 7181–7190.

[9] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Know your surroundings: Exploiting scene information for object tracking," in *ECCV*. Springer, 2020, pp. 205–221.

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-Convolutional Siamese Networks for Object Tracking," in *ECCV*, 2016, pp. 850–865.

[11] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High Performance Visual Tracking With Siamese Region Proposal Network," in *CVPR*, 2018, pp. 8971–8980.

[12] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese Box Adaptive Network for Visual Tracking," in *CVPR*, 2020, pp. 6667–6676.

[13] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning Spatio-Temporal Transformer for Visual Tracking," in *ICCV*, 2021, pp. 10 428–10 437.

[14] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer Tracking," in *CVPR*, 2021, pp. 8126–8135.

[15] B. Ye, H. Chang, B. Ma, S. Shan, and X. Chen, "Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework," in *ECCV*, 2022, pp. 341–357.

[16] X. Chen, H. Peng, D. Wang, H. Lu, and H. Hu, "Seqtrack: Sequence to sequence learning for visual object tracking," in *CVPR*, 2023, pp. 14 572–14 581.

[17] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines." in *AAAI*, 2020, pp. 12 549–12 556.

[18] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking," in *CVPR*, 2020, pp. 6268–6276.

[19] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks," in *CVPR*, 2019, pp. 4282–4291.

[20] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast Online Object Tracking and Segmentation: A Unifying Approach," in *CVPR*, 2019, pp. 1328–1338.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[22] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "Hift: Hierarchical feature transformer for aerial tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 457–15 466.

[23] L. Yao, C. Fu, S. Li, G. Zheng, and J. Ye, "SGDViT: saliency-guided dynamic vision transformer for uav tracking," in *ICRA*. IEEE, 2023, pp. 3353–3359.

[24] C. Fu, M. Cai, S. Li, K. Lu, H. Zuo, and C. Liu, "Continuity-aware latent interframe information mining for reliable uav tracking," in *ICRA*. IEEE, 2023, pp. 1327–1333.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *ICLR*, 2021.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," in *ICML*, 2021, pp. 8748–8763.

[27] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, and J. Wang, "Context autoencoder for self-supervised representation learning," *International Journal of Computer Vision*, pp. 1–16, 2023.

[28] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021.

[29] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[30] B. Chen, P. Li, L. Bai, L. Qiao, Q. Shen, B. Li, W. Gan, W. Wu, and W. Ouyang, "Backbone is All Your Need: A Simplified Architecture for Visual Object Tracking," in *ECCV*, 2022, pp. 375–392.

[31] Y. Cai, J. Liu, J. Tang, and G. Wu, "Robust object modeling for visual tracking," in *ICCV*, 2023, pp. 9589–9600.

[32] B. Kang, X. Chen, D. Wang, H. Peng, and H. Lu, "Exploring lightweight hierarchical vision transformers for efficient visual tracking," in *ICCV*, 2023.

[33] Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye, "Conformer: Local features coupling global representations for visual recognition," in *ICCV*, 2021, pp. 367–376.

[34] M. Mueller, N. Smith, and B. Ghanem, "A Benchmark and Simulator for UAV Tracking," in *ECCV*, 2016, pp. 445–461.

[35] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[36] S. Gao, C. Zhou, and J. Zhang, "Generalized relation modeling for transformer tracking," in *CVPR*, 2023, pp. 18 686–18 695.

[37] J. Zhu, S. Lai, X. Chen, D. Wang, and H. Lu, "Visual prompt multi-modal tracking," in *CVPR*, 2023, pp. 9516–9526.

[38] W. Cai, Q. Liu, and Y. Wang, "Hiptrack: Visual tracking with historical prompts," in *CVPR*, 2024, pp. 19 258–19 267.

[39] X. Ma, X. Dai, Y. Bai, Y. Wang, and Y. Fu, "Rewrite the stars," in *CVPR*, 2024.

[40] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *ICCV*, 2019, pp. 4471–4480.

[41] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "Siamapn++: Siamese attentional aggregation network for real-time uav tracking," in *IROS*. IEEE, 2021, pp. 3086–3092.

[42] Z. Cao, Z. Huang, L. Pan, S. Zhang, Z. Liu, and C. Fu, "Tctrack: Temporal contexts for aerial tracking," in *CVPR*, 2022, pp. 14 798–14 808.

[43] ——, "Towards real-world visual tracking with temporal contexts," *IEEE TPAMI*, 2023.

[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[45] S. Li, Z. Wang, Z. Liu, C. Tan, H. Lin, D. Wu, Z. Chen, J. Zheng, and S. Z. Li, "Moganet: Multi-order gated aggregation network," in *ICLR*, 2022.

[46] X. Wei, Y. Bai, Y. Zheng, D. Shi, and Y. Gong, "Autoregressive visual tracking," in *CVPR*, June 2023, pp. 9697–9706.

[47] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze, "LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference," in *ICCV*, 2021, pp. 12 239–12 249.

[48] C. Fu, Z. Cao, Y. Li, J. Ye, and C. Feng, "Onboard real-time aerial tracking with efficient siamese anchor proposal network," *IEEE TGARS*, vol. 60, pp. 1–13, 2021.

[49] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.

[50] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *CVPR*, 2019, pp. 658–666.

[51] L. Huang, X. Zhao, and K. Huang, "Got-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild," *IEEE TPAMI*, vol. 43, no. 5, pp. 1562–1577, 2021.

[52] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking," in *CVPR*, 2019, pp. 5374–5383.

[53] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick,

"Microsoft COCO: Common Objects in Context," in *ECCV*, 2014, pp. 740–755.

[54] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild," in *ECCV*, 2018, pp. 310–327.

[55] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *ICLR*, 2019.

[56] H. Zuo, C. Fu, S. Li, J. Ye, and G. Zheng, "End-to-end feature decontaminated network for uav tracking," in *IROS*. IEEE, 2022, pp. 12 130–12 137.

[57] C. Fu, W. Peng, S. Li, J. Ye, and Z. Cao, "Local perception-aware transformer for aerial tracking," in *IROS*. IEEE, 2022, pp. 12 122–12 129.

[58] H. Zuo, C. Fu, S. Li, J. Ye, and G. Zheng, "Deconnet: End-to-end decontaminated network for vision-based aerial tracking," *IEEE TGARS*, vol. 60, pp. 1–12, 2022.

[59] D. Xing, N. Evangeliou, A. Tsoukalas, and A. Tzes, "Siamese transformer pyramid networks for real-time uav tracking," in *WACV*, 2022, pp. 2139–2148.

[60] Z. Zhang and H. Peng, "Deeper and Wider Siamese Networks for Real-Time Visual Tracking," in *CVPR*, 2019, pp. 4591–4600.

[61] I. Sosnovik, A. Moskalev, and A. W. Smeulders, "Scale equivariance improves siamese tracking," in *WACV*, 2021, pp. 2765–2774.

[62] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware Anchor-free Tracking," in *ECCV*, 2020, pp. 771–787.