

SiamAPN++: Siamese Attentional Aggregation Network for Real-Time UAV Tracking

Ziang Cao¹, Changhong Fu^{2,*}, Junjie Ye², Bowen Li², and Yiming Li³

Abstract—Recently, the Siamese-based method has stood out from multitudinous tracking methods owing to its state-of-the-art (SOTA) performance. Nevertheless, due to various special challenges in UAV tracking, *e.g.*, severe occlusion and fast motion, most existing Siamese-based trackers hardly combine superior performance with high efficiency. To this concern, in this paper, a novel attentional Siamese tracker (SiamAPN++) is proposed for real-time UAV tracking. By virtue of the attention mechanism, we conduct a special attentional aggregation network (AAN) consisting of self-AAN and cross-AAN for raising the representation ability of features eventually. The former AAN aggregates and models the self-semantic interdependencies of the single feature map via spatial and channel dimensions. The latter aims to aggregate the cross-interdependencies of two different semantic features including the location information of anchors. In addition, the anchor proposal network based on dual features is proposed to raise its robustness of tracking objects with various scales. Experiments on two well-known authoritative benchmarks are conducted, where SiamAPN++ outperforms its baseline SiamAPN and other SOTA trackers. Besides, real-world tests onboard a typical embedded platform demonstrate that SiamAPN++ achieves promising tracking results with real-time speed.

I. INTRODUCTION

Visual object tracking is a fundamental and challenging task, whose purpose is to track the indicated object frame by frame. By virtue of the powerful flexibility of unmanned aerial vehicles (UAVs), UAV tracking has drawn considerable attention in many fields such as aerial cinematography [1], path planning [2], and self-localization [3]. Despite great efforts, designing an efficient, accurate, and robust tracker for UAV remains an extremely challenging task. On the one hand, the limited computational resource on the embedded platform hardly meets the requirement of existing robust but computation-consuming methods. On the other hand, the UAV tracking also suffers from various special challenges brought by the mobile platform, *e.g.*, fast motion, low resolution, and severe occlusion.

Generally, there are two mainstream methods in the field of UAV tracking, *i.e.*, correlation filter (CF)-based method and deep learning (DL)-based method. According to [4], online CF-based trackers are widely adopted in UAV tracking due to their low computational complexity [5], [6]. In spite

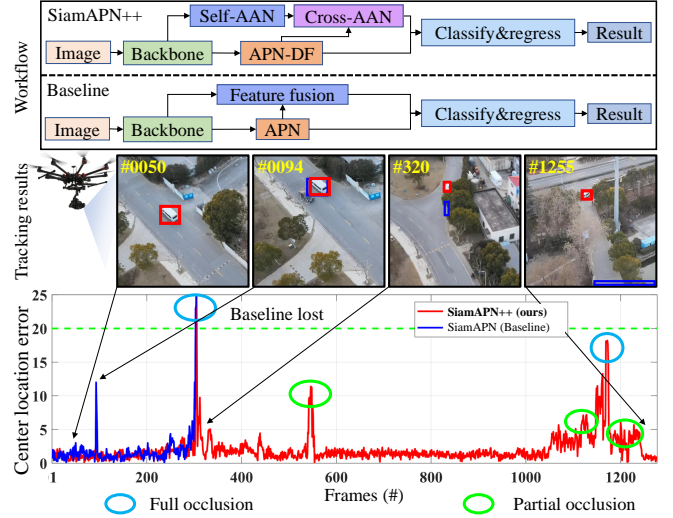


Fig. 1. Comparison between the baseline tracker and the proposed tracker, *i.e.*, SiamAPN and SiamAPN++. The figures from the top to bottom are workflow, tracking results, and center location error comparison. When facing the constantly occluded object, the attentional aggregation network (AAN) can emphasize and aggregate the effective information selectively, avoiding interference of environment.

of high efficiency, the accuracy and robustness of CF-based trackers fail to meet the requirement of practical UAV tracking in complex scenes. Meantime, albeit the DL-based methods have also achieved significant advancement in tracking performance, the large computation cost limits their practical application. Consequently, it is urgent to develop an efficient DL-based structure to balance performance and speed.

Among the DL-based trackers, the Siamese-based trackers enjoy their superiority due to their high potential in object tracking. The Siamese network structure is widely spread by SiamFC [7]. It proposed a new tracking strategy, *i.e.*, tracking by computing the similarity between template and search patches. Then, the anchor-based method is proposed in SiamRPN [8]. By introducing the region proposal network (RPN), it can obtain accurate bounding boxes. Based on RPN, further researches have been made to improve the tracking performance [9]–[11]. Since those anchors are predefined, the performance of anchor-based trackers is seriously influenced by the hyper-parameters associated with anchors. To improve the generalization of trackers, the anchor-free method is proposed, which generally predicts the offset between ground truth and center point [12]–[14]. Although the anchor-free method avoids the hyper-parameters, the phe-

*Corresponding Author

¹Ziang Cao is with the School of Automotive Studies, Tongji University, 201804 Shanghai, China.

²Changhong Fu, Junjie Ye, and Bowen Li are with the School of Mechanical Engineering, Tongji University, 201804 Shanghai, China. changhongfu@tongji.edu.cn

³Yiming Li is with the Tandon School of Engineering, New York University, NY 11201 New York, United States.

nomenon of imbalanced samples is still not properly solved. To solve this problem, a new anchor-free method is raised by the anchor proposal network (APN) [15]. However, it is not good enough to handle semantic information variation.

Meantime, the attention mechanism has also drawn much attraction in object tracking [11], [16]. Unfortunately, the relationship and interdependencies of different features are generally neglected, impeding the robustness under various conditions. Therefore, we propose a novel attentional Siamese tracker namely SiamAPN++ for real-time UAV tracking, mainly consisting of the attentional aggregation network (AAN) and anchor proposal network based on dual features (APN-DF) illustrated in Fig. 1. The AAN is divided into two subnetworks, *i.e.*, self-AAN and cross-AAN. The former aims to model the self-semantic interdependencies in a single feature map via spatial and channel dimensions while the latter focuses on emphasizing the interdependent channels of different feature maps adaptively. Besides, to promote the robustness of tracking objects with various scales, dual features are introduced in APN. By exploiting the interdependencies of features from different levels, APN-DF can generate more appropriate anchors than before. Real-world tests onboard the embedded platform shown in Fig. 1 strongly prove the superior accuracy and robustness of SiamAPN++ while keeping a comparable speed to SiamAPN.

The main contributions of this work can be summarized as follows:

- A novel AAN is introduced to aggregate the self-semantic interdependencies of the single feature map via two mechanisms and the cross-interdependencies from different feature maps adaptively.
- The dual features are adopted in APN-DF, thereby improving the anti-interference and robustness of the proposed anchors when facing severe scale variation.
- Extensive evaluations on two challenging UAV tracking benchmarks prove the superior performance of SiamAPN++ especially in fast motion, low resolution, and severe occlusion. In addition, real-world tests conducted onboard an embedded platform strongly demonstrate impressive practicability and performance of our tracker with real-time speed.

II. RELATED WORKS

Previously, CF-based trackers have attracted much attention since MOSSE [17]. Depending on the high efficiency and expansibility, CF-based trackers can be deployed directly on UAVs. There is no doubt that the CF-based approaches promoted the development of UAV tracking with satisfying speed [6], [18]. However, the online tracking strategy hinders inevitably the structure of trackers, hardly maintaining satisfying tracking performance in practical conditions.

The Siamese-based network also shows its huge potential in the domain of object tracking. SINT [19] firstly view the tracking task as matching the patch problem. Since the appearance of SiamFC [7], the advantage of the Siamese network has been obvious. It aims to measure the similarity be-

tween template and search patches by employing a fully convolutional neural network. Then, SiamRPN [20] introduced the RPN into the Siamese framework, dividing the tracking task into classification and regression. DaSiamRPN [10] proposes a novel training method, further improving the tracking performance. Moreover, SiamRPN++ [9] makes it possible to utilize deeper networks as backbone. Despite obtaining state-of-the-art (SOTA) performance, those anchor-based trackers above suffer from hyper-parameters and imbalanced samples. In order to eliminate these problems, anchor-free trackers are proposed, *e.g.*, SiamFC++ [12] and SiamCAR [13]. By redesigning the regression, the generalization of the trackers is raised. However, the influence caused by imbalanced samples is still existing. SiamAPN [21] brings a novel approach to handle the two problems at the same time, *i.e.*, proposing adaptive anchors. It increases the proportion of positive samples while adopting the no-prior structure of APN.

In recent, much attention has been paid to the attention mechanism in many fields. For capturing long-range dependencies, the non-local block is proposed in [22]. Then, DANet [23] develops the self-attention mechanism to address the scene segmentation task. In visual tracking, RASNet [16] adopts three attention branches to adapt the model without updating the model online in visual tracking. SiamAttn [11] also exploits the attention mechanism module for providing an implicit manner to update the template. Besides, ECA-Net [24] proposes an efficient channel attention method for the deep convolutional neural network. However, those methods mentioned above merely focus on self-attention or the relationship between template and searches, neglecting the cross-interdependence between different level features.

In this work, the cross-AAN is proposed for exploring the potential of the cross-semantic interdependencies contained in different level features. Plus the self-interdependencies of the single feature map, the AAN can raise the representation ability of features effectively for handling the special challenges in UAV tracking. Besides, the APN based on dual feature structure is reconstructed for raising the anti-interference and robustness of proposing anchors and provide the reliable internal feature for AAN.

III. METHODOLOGY

A. Revisit SiamAPN

In this section, our baseline is revisited in brief, consisting of four subnetworks, *i.e.*, feature extraction network, APN, feature fusion network, and classification®ression network. Different from pre-defined anchors, the APN adopts a single feature map to generate adaptive anchors, avoiding the hyper-parameters associated with those pre-defined anchors. Besides, it also decreases the number of negative samples, alleviating the phenomenon of imbalanced samples.

Although SiamAPN has achieved competitive performance, it remains two shortcomings: *a)* the performance is influenced easily by complex semantic information variation. *b)* the original APN hardly maintains satisfying robustness when facing objects with various scales.

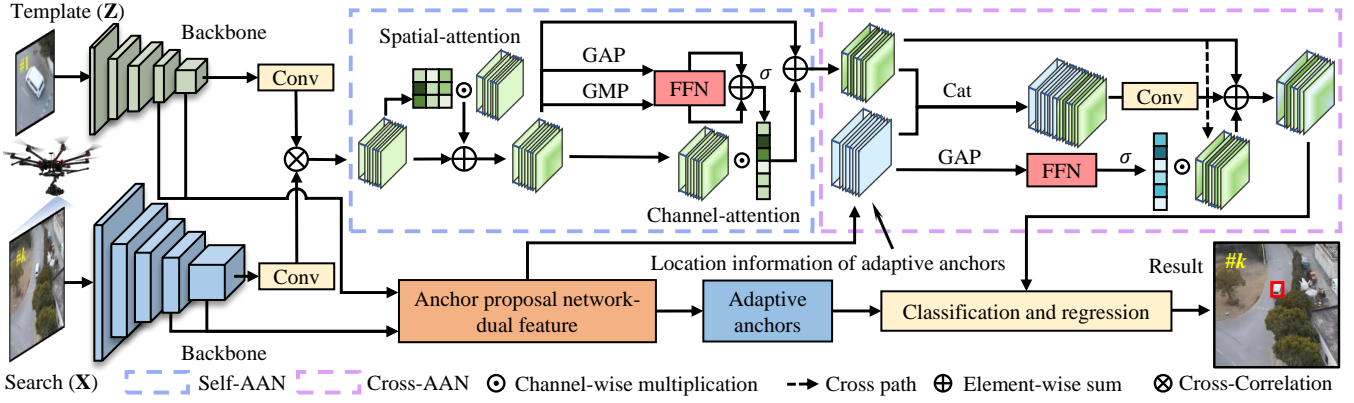


Fig. 2. The overview of the SiamAPN++ tracker. It composes of four subnetworks, *i.e.*, feature extraction network (backbone), classification and regression network, anchor proposal network-dual feature (APN-DF), and attentional aggregation network (AAN). (Best viewed in color version)

B. Proposed method

In this section, the proposed SiamAPN++ is introduced in detail. As shown in Fig. 2, SiamAPN++ consists of four subnetworks, *i.e.*, the feature extraction network, APN-DF, AAN, and classification®ression network.

1) *Feature extraction network*: For fulfilling the practical deployment requirement, AlexNet [25] is chosen as the backbone of the SiamAPN++ tracker. In our tracker, the feature maps from the last two layers are utilized in the tracking task. For clarification, the template image, search image, and the output of the k^{th} layer are denoted as \mathbf{Z} , \mathbf{X} , and $\varphi_k(\cdot)$ respectively.

2) *APN-DF*: For the sake of boosting the robustness of the proposed anchors, the dual features from the feature extraction network are adopted to handling objects with different scales. Generally, the high-level features are concentrate on semantic features which are good for classifying while the low-level ones focus on detailed features which are helpful for accurately distinguishing objects. Therefore, we design a new APN structure to discover the internal relationship between the two different feature maps. In this way, we combine the dual features while avoiding the extra computation brought by separately calculating for each feature map.

Specifically, the $\varphi_4(\mathbf{X})$ and $\varphi_4(\mathbf{Z})$ is used to produce the similarity map as:

$$\mathbf{R}_4 = \mathcal{F}(\varphi_4(\mathbf{X}) \star \varphi_4(\mathbf{Z})) , \quad (1)$$

where the \star represents the depth-wise cross-correlation layer introduced in [9] and \mathcal{F} denotes the convolutional operation for decreasing the number of channels. Additionally, the similarity of the fifth layer is acquired as:

$$\mathbf{R}_5 = \mathcal{F}(\varphi_5(\mathbf{X})) \star \mathcal{F}(\varphi_5(\mathbf{Z})) . \quad (2)$$

Then, the feature maps $\mathbf{R}_A \in \mathbb{R}^{C \times H \times W}$ can be formulated as:

$$\begin{aligned} \mathbf{R}_A = & \mathbf{R}_5 + \gamma_1 \star \text{FFN}(\text{GAP}(\mathbf{R}_4)) \star \mathbf{R}_5 \\ & + \gamma_2 \star \mathcal{F}(\text{Cat}(\mathbf{R}_4, \mathbf{R}_5)) , \end{aligned} \quad (3)$$

where γ_k , $k = \{1, 2\}$, GAP, and Cat represent the learning weights, global average pooling, and channel-wise concatenation respectively. After obtaining the \mathbf{R}_A , the anchors can be obtained through two convolutional operations.

Remark 1: Since the dual feature structure exploits and integrates the preponderance of different level features, the robustness and anti-interference ability under scale variation are raised. Besides, it can also provide comprehensive position information for AAN.

3) *AAN*: AAN consists of self-AAN and cross-AAN, designing for effectively improving the representation ability of feature maps via adaptively enhancing self-semantic interdependencies and aggregating the cross-interdependencies.

Self-AAN: The self-AAN aims to enhance the self-semantic information of the single feature map via spatial and channel attention. The spatial attention is inspired by [23]. To explore different semantic information, we use different convolution layer to calculate \mathbf{R}'_5 as follow:

$$\mathbf{R}'_5 = \mathcal{F}(\varphi_5(\mathbf{X})) \star \mathcal{F}(\varphi_5(\mathbf{Z})) . \quad (4)$$

Then, by operating three different convolution layers, three new feature maps can be generated denoted as $\{\mathbf{R}^q, \mathbf{R}^k, \mathbf{R}^v\} \in \mathbb{R}^{C \times H \times W}$. By reshaping \mathbf{R}^q and \mathbf{R}^k to $\mathbb{R}^{C \times (H \times W)}$, cooperating matrix multiplication, and softmax layer, the spatial attention map $\mathbf{R}^a \in \mathbb{R}^{(H \times W) \times (H \times W)}$ can be calculated. Therefore, the output of spatial attention $\mathbf{R}^s \in \mathbb{R}^{C \times H \times W}$ can be obtained as:

$$\mathbf{R}^s = \gamma_3 \mathbf{R}^v \times \mathbf{R}^a + \mathbf{R}'_5 , \quad (5)$$

where γ_3 is a learning weight and \times denotes the matrix multiplication.

For exploiting the interdependencies between channels and improve the representation ability of features, we also build the channel-attention branch. To obtain exhaustive channel enhancement, global max-pooling (GMP) and GAP are adopted before FFN. Besides, the channel attention FFN is share-weight to explore the interdependencies among channels more effectively. Therefore, the calculation process of $\mathbf{R}_c \in \mathbb{R}^{C \times H \times W}$ is as follows:

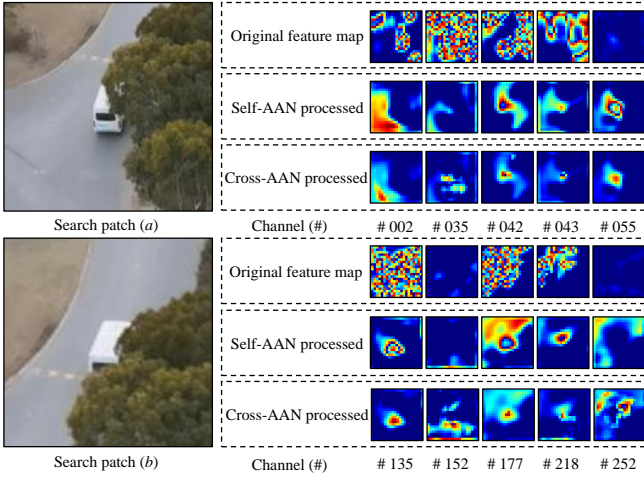


Fig. 3. Visualization of the AAN output feature maps in the real-world test. It shows that AAN indeed provides effective and robust feature maps for tracking occluded objects accurately.

$$\begin{aligned} \mathbf{w} &= \text{FFN}(\text{GAP}(\mathbf{R}^s)) + \text{FFN}(\text{GMP}(\mathbf{R}^s)) \\ \mathbf{R}_c &= \mathbf{R}^s + \gamma_4 \text{Sigmoid}(\mathbf{w}) * \mathbf{R}^s, \end{aligned} \quad (6)$$

where γ_4 is a learning weight and $*$ denotes the channel-wise multiplication.

Remark 2: By building the spatial and channel attention, the self-AAN can aggregate and model the self-semantic interdependencies of the single feature map. Therefore, it can provide stable and robust self-attentional features for cross-AAN.

Cross-AAN: Since the anchors are generated adaptively, semantic information about anchors is necessary for SiamAPN++ to locate the anchors before predicting. Therefore, the location information of anchors is quite significant for classification and regression performance. To aggregate the cross-interdependencies between two different features and integrate the location information of anchors, cross-AAN is introduced.

There are two cross paths in cross-AAN. One path applies a feedforward neural network (FFN) structure to generate a corresponding weight for each channel of \mathbf{R}_c , aggregating the interdependencies among channels explicitly. Besides, before the FFN, GAP is cooperated to compress the feature to vector. The other path aims to emphasize the interdependent channels of \mathbf{R}_A and \mathbf{R}_c implicitly via element-wise concatenation. Therefore, the final feature maps $\mathbf{R} \in \mathbb{R}^{C \times H \times W}$ can be computed as:

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_c + \gamma_5 * \text{FFN}(\text{GAP}(\mathbf{R}_A)) * \mathbf{R}_c \\ &\quad + \gamma_6 * \mathcal{F}(\text{Cat}(\mathbf{R}_A, \mathbf{R}_c)), \end{aligned} \quad (7)$$

where γ_k , $k = \{5, 6\}$ are also learning weights to adaptively maintain the balance between the different branches.

Remark 3: By virtue of cross-ANN, the cross-interdependencies between \mathbf{R}_A and \mathbf{R}_c are aggregated. Figure 3 clearly shows that AAN adaptively highlights the effective information from complex feature maps and effectively weakens the interference factors caused by occlusion. Eventually, based on the improved feature map, the robustness and accuracy of SiamAPN++ are raised.

4) *Classification and regression network:* The classification and regression network apply a similar structure compared with the baseline, *i.e.*, SiamAPN. The multi-classification structure is also adopted. The first branch aims to classify the anchors with a high intersection over union (IoU) score. The second branch concentrates on selecting the points on the feature map that fall in ground truth areas. The last branch considers the center distance between each point and ground truth center point inspired by [13]. w_1 , w_2 , and w_3 are introduced for balancing different branches. For improving the performance of convergence, the loss function of regression is redesigned as:

$$L_{\text{ious}} = -(1 - \text{ious}) * (\alpha - \text{ious}) * \log(\text{ious}), \quad (8)$$

where α is a hyper-parameter, reflecting the tendentiousness to positive and negative samples while ious represents the IoU score between the proposed anchors and ground truth bounding box. α is set in the range of 1 to 2. It aims to increase gradient at high loss position to accelerate the convergence process, and decrease gradient at low loss position to raise the convergence accuracy. In this way, the model obtains accurate and robust convergence results faster (epoch=25) compared with the baseline (epoch=37).

IV. EVALUATIONS

In this section, SiamAPN++ is comprehensively evaluated on two well-known authoritative UAV tracking benchmarks, *i.e.*, UAV20L [26] and UAV123@10fps [26]. Other 14 SOTA trackers are also included in the evaluation, *i.e.*, SiamAPN (baseline) [15], SiamRPN++ [9], DaSiamRPN [10], SiamFC++ [12], SiamFC [7], UDT [27], UDT+ [27], TADT [28], DSiam [29], CoKCF [30], CF2 [31], CFNet [32], ECO [33], and KCC [34].

Remark 4: To better compare the performance of different tracking strategies, all Siamese-based trackers adopt the same backbone, *i.e.*, AlexNet [25] pre-trained on ImageNet [35].

A. Implementation details

In the training process, the last three layers of the backbone are fine-tuned with a learning rate of 5×10^{-4} at first. The whole tracker are trained on the images extracted from COCO [36], ImageNet VID [35], GOT-10K [37] and Youtube-BB [38]. The stochastic gradient descent (SGD) with a minibatch of 220 pairs is applied. Besides, the momentum is set to 0.9 and weight decay is set to 10^{-4} . Following the baseline, the input size of the template image and search image are set to $3 \times 127 \times 127$ pixels and $3 \times 287 \times 287$ pixels. The tracking code is available at <https://github.com/vision4robotics/SiamAPN>.

The training process of the proposed method SiamAPN++ is implemented in Python using Pytorch on a PC with an Intel i9-9920X CPU, a 32GB RAM, and two NVIDIA TITAN RTX GPUs. For testing the feasibility and performance of SiamAPN++ on UAV tracking, an NVIDIA Jetson AGX Xavier is adopted as the real-world tests platform. Real-world tests validate the accuracy and robustness of SiamAPN++ with a speed of around 35 frames per second (FPS) without TensorRT acceleration.

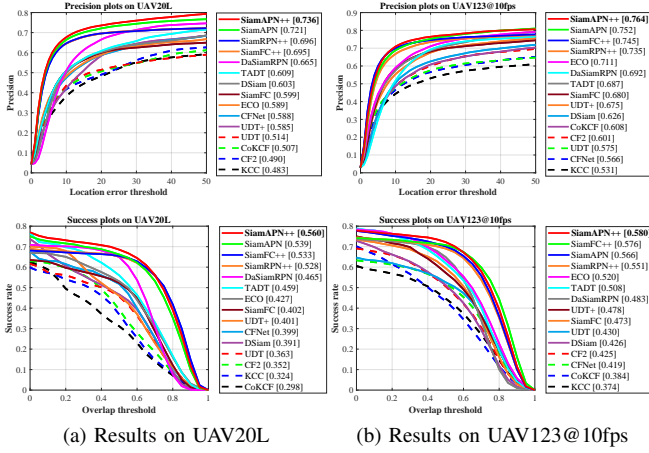


Fig. 4. Overall performance of all trackers on (a) UAV20L, (b) UAV123@10fps. The overall results illustrate that SiamAPN++ achieves superior performance against other SOTA trackers.

TABLE I

AVERAGE ATTRIBUTE-BASED EVALUATION OF THE SIAMAPN++ AND OTHER 14 SOTA TRACKERS ON TWO BENCHMARKS. THE BEST THREE PERFORMANCES ARE RESPECTIVELY HIGHLIGHTED WITH RED, GREEN, AND BLUE COLOR.

	CM		FM		FOC		POC		SV	
Trackers	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
ECO	0.630	0.457	0.542	0.344	0.457	0.254	0.598	0.427	0.621	0.453
SiamFC	0.626	0.432	0.529	0.319	0.487	0.264	0.577	0.385	0.614	0.416
UDT	0.512	0.375	0.470	0.303	0.413	0.226	0.493	0.350	0.513	0.375
UDT+	0.598	0.424	0.521	0.322	0.469	0.259	0.581	0.403	0.599	0.418
CF2	0.519	0.374	0.384	0.246	0.431	0.233	0.499	0.346	0.507	0.364
CFNet	0.533	0.382	0.435	0.244	0.407	0.214	0.527	0.365	0.542	0.386
CoKCF	0.525	0.329	0.414	0.210	0.418	0.205	0.512	0.309	0.520	0.311
DSiam	0.613	0.404	0.536	0.309	0.547	0.301	0.579	0.371	0.585	0.387
TADT	0.638	0.474	0.597	0.386	0.486	0.281	0.612	0.444	0.618	0.461
KCC	0.458	0.320	0.348	0.206	0.339	0.172	0.457	0.300	0.467	0.320
DaSiamRPN	0.667	0.470	0.571	0.365	0.505	0.274	0.625	0.427	0.656	0.459
SiamFC++	0.709	0.544	0.622	0.454	0.469	0.292	0.669	0.497	0.700	0.540
SiamRPN++	0.706	0.530	0.624	0.442	0.475	0.284	0.655	0.480	0.691	0.521
SiamAPN	0.721	0.537	0.710	0.493	0.520	0.308	0.680	0.493	0.715	0.537
SiamAPN++	0.742	0.564	0.711	0.507	0.577	0.358	0.702	0.519	0.729	0.554

B. Evaluation metrics

The one-pass evaluation (OPE) metrics [39] are adopted, *i.e.*, precision and success rate. Specifically, the success rate is measured by the IoU score. Besides, the success plot reflects the percentage of the frames whose IoU score is beyond a pre-defined threshold and the area under the curve (AUC) of success plot is used to rank all the trackers. The precision plot shows the percentage of frames whose center location error (CLE) between the estimated bounding box and ground truth is smaller than thresholds. Note that the score at 20 pixels is utilized for ranking.

C. Evaluation on UAV benchmarks

1) *Overall performance*: The proposed method achieves an impressive improvement compared with other SOTA trackers on two well-known benchmarks.

UAV20L: UAV20L [26] contains 20 long-term sequences whose maximum sequence contains 5527 frames with an average of 2934 frames per sequence. Therefore, UAV20L is adopted for evaluating long-term tracking performance. As illustrated in Fig. 4a, SiamAPN++ outperforms other trackers with an improvement of 2.0% on precision and 4.0% on

TABLE II
PRECISION COMPARISON OF OUR TRACKER WITH DIFFERENT COMPONENTS ON UAV20L. NOTE THAT ARC REPRESENTS THE ASPECT RATIO CHANGE AND OC INCLUDES FULL OCCLUSION AS WELL AS PARTIAL OCCLUSION.

Structure	Overall	SV	ARC	CM	OC
Baseline	0.721	0.707	0.652	0.707	0.617
Baseline+APN-DF	0.727	0.715	0.662	0.713	0.649
Baseline+APN-DF+self-AAN	0.730	0.716	0.663	0.716	0.647
Baseline+APN-DF+AAN	0.736	0.722	0.670	0.722	0.651

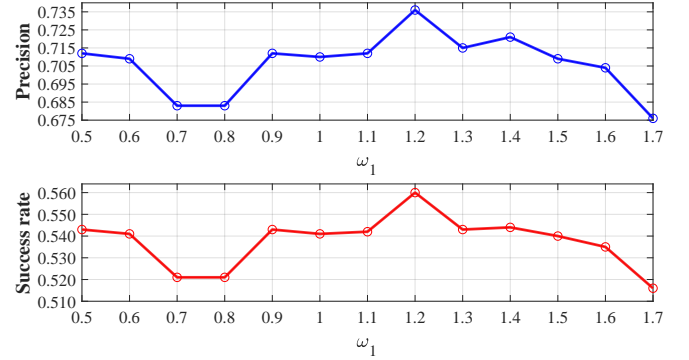


Fig. 5. Key parameter analysis of w_1 on UAV20L. When the $w_1 = 1.2$, SiamAPN++ achieves the best overall performance, which is adopted in all experiments

AUC score, compared with the second-best tracker.

UAV123@10fps: UAV123@10fps [26] contains 123 sequences with a frame rate of 10 FPS. Since the frame interval becomes larger than 30 FPS, the movement and variation of the object become more drastic, bringing difficulties to the tracking task. Therefore, UAV123@10fps [26] is chosen to comprehensively evaluate the robustness of the tracker under severe variation. Overall performance shown in Fig. 4b proves the superior robustness and accuracy of SiamAPN++ in precision (0.764) and AUC score (0.580).

2) *Attribute-based performance*: To analyze the robustness of SiamAPN++ under various challenges, the average

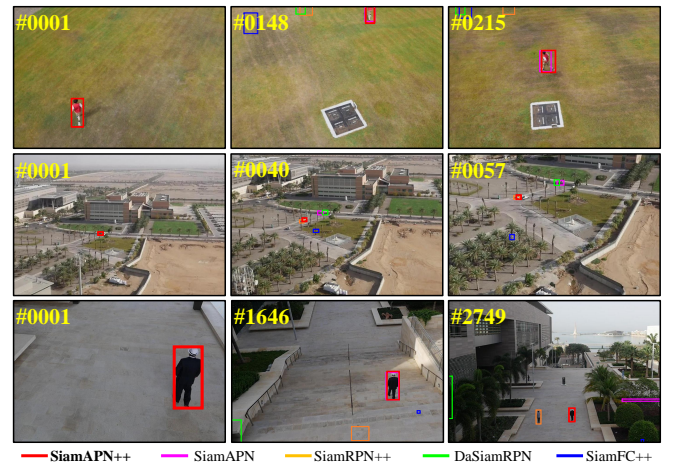


Fig. 6. Screenshots of *person7_2*, *car11* from UAV123@10fps, and *person19* from UAV20L. The tracking videos can be found here: <https://youtu.be/okS289p3pCQ>.

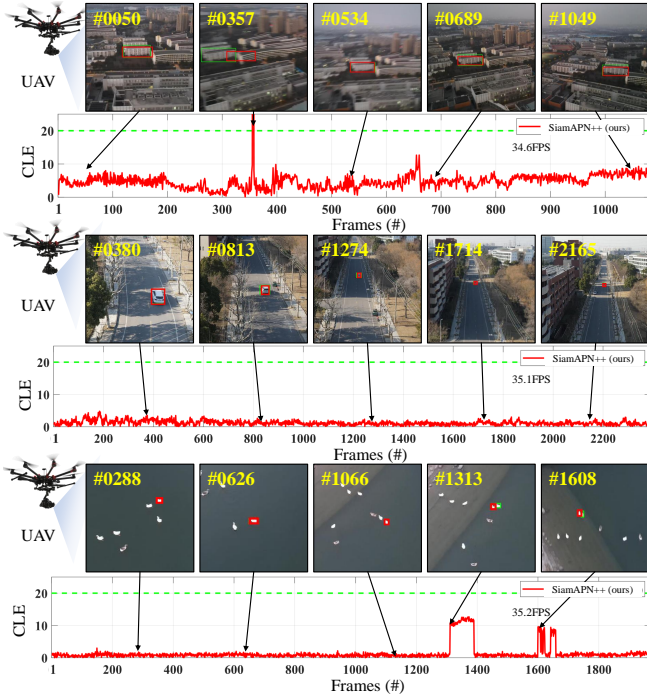


Fig. 7. Real-world tests in terms of CLE onboard the embedded platform. The tracking objects from the top to bottom are building, car, and duck. Besides, the tracking results and ground truth are marked with red and green boxes. The CLE score below the green dotted line is considered as the effective tracking result.

attribute-based evaluation results on two UAV benchmarks are shown in TABLE I. Five most common attributes in UAV tracking challenges are analyzed, *i.e.*, camera motion (CM), fast motion (FM), full occlusion (FOC), partial occlusion (POC), and scale variation (SV). Attributing to the AAN and dual feature structure, SiamAPN++ achieves impressive performance in the CM scenarios with a **5.0%** promotion on AUC score. Besides, our tracker surpasses the baseline in terms of FM. Meantime, in the FOC conditions, SiamAPN++ exceeds the second-best tracker with a huge improvement of **5.5%** in precision and **16.2%** in AUC score.

3) *Ablation study*: To demonstrate the effectiveness of the APN-DF and AAN, the precision of SiamAPN++ with different components and the baseline SiamAPN on UAV20L is listed in TABLE II. With the internal relationship introduced by the APN-DF, the tracker has surpassed the baseline. Besides, it indeed promotes performance when tracking objects with various scales. Furthermore, attributing to the AAN, the self-interdependencies from the single feature map and the cross-interdependencies are aggregated, further improving the accuracy of SiamAPN++.

4) *Key parameter analysis*: Since the first classification branch, *i.e.*, w_1 , directly reflects the classification tendentiousness to anchors, w_1 has important influence in tracking performance. Obviously, too large w_1 will influence the effectiveness of other branches, while too small w_1 will impede the promotion introduced by anchor information on classification. Therefore, to evaluate the influence of w_1 , w_1 is set to different values for further research. It is set from 0.5 and increases in a small step of 0.1. As presented in

Fig. 5, the AUC and precision of SiamAPN++ achieve best performance when $w_1 = 1.2$. Please note that w_1 is set to 1.2 during the evaluation above and the real-world tests.

5) *Qualitative Evaluation*: Some qualitative comparisons are shown in Fig. 6. The main challenges of these sequences include SV, ARC, FM, POC, CM, and out-of-view. By the combination of APN-DF and AAN, SiamAPN++ achieves superior tracking performance eventually.

V. REAL-WORLD TESTS

To verify the feasibility of the proposed tracker, real-world tests are conducted onboard the embedded platform. Four real-world tests are presented in Fig. 7 and Fig. 1 includes car, building, and duck. The main challenges in the first test illustrated in Fig. 7 are severe CM, POC, and similar objects. Although there are some errors due to the blurring caused by severe CM, SiamAPN++ can determine the location of the object again and achieve impressive long-term tracking performance. By aggregating self- and cross-interdependencies of feature maps, SiamAPN++ can handle the low resolution (LR), SV, ARC challenges effectively, thereby tracking the car accurately and robustly in the second scene. Meantime, benefiting from the location information of anchors, the robustness of SiamAPN++ is also improved. Based on it, the AAN can explore the effective interdependencies between feature maps, thereby achieving impressive tracking performance in the third test. Additionally, the test illustrated in Fig. 1 mainly focuses on the occluded object, validating the effectiveness of the combination of AAN and APN-DF when facing severe occlusion. During the real-world tests, our tracker maintains robust and accurate performance with an average speed of 34.9 FPS. In a word, the real-world tests strongly prove the impressive performance of SiamAPN++ under various challenges with a promising speed in UAV tracking scenarios.

VI. CONCLUSION

In this work, a novel attentional Siamese-based tracker is introduced for fulfilling the performance and feasibility requirement of real-time UAV tracking. The self-AAN is proposed for aggregating the self-interdependencies of the single feature. Besides, to aggregate the cross-interdependencies between two different feature maps, we also propose the cross-AAN. In addition, the new dual feature structure also integrates different feature maps effectively. Exhaustive evaluations validate the effectiveness of our method. Meantime, real-world tests strongly verify the practicability of our tracker. Consequently, we believe that our work can boost the development of UAV tracking-related applications.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61806148) and the Natural Science Foundation of Shanghai (No. 20ZR1460100).

REFERENCES

- [1] R. Bonatti, C. Ho, W. Wang, S. Choudhury, and S. Scherer, "Towards a Robust Aerial Cinematography Platform: Localizing and Tracking Moving Targets in Unstructured Environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 229–236.
- [2] G. J. Laguna and S. Bhattacharya, "Path planning with Incremental Roadmap Update for Visibility-based Target Tracking," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1159–1164.
- [3] J. Ye, C. Fu, F. Lin, F. Ding, S. An, and G. Lu, "Multi-Regularized Correlation Filter for UAV Tracking and Self-Localization," *IEEE Transactions on Industrial Electronics*, pp. 1–10, 2021.
- [4] C. Fu, B. Li, F. Ding, F. Lin, and G. Lu, "Correlation Filter for UAV-Based Aerial Tracking: A Review and Experimental Evaluation," *IEEE Geoscience and Remote Sensing Magazine*, pp. 1–28, 2020.
- [5] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning Aberrance Repressed Correlation Filters for Real-time UAV Tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 2891–2900.
- [6] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "AutoTrack: Towards High-Performance Visual Tracking for UAV With Automatic Spatio-Temporal Regularization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 920–11 929.
- [7] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-Convolutional Siamese Networks for Object Tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 850–865.
- [8] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High Performance Visual Tracking with Siamese Region Proposal Network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8971–8980.
- [9] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4277–4286.
- [10] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-Aware Siamese Networks for Visual Object Tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 101–117.
- [11] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable Siamese Attention Networks for Visual Object Tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6727–6736.
- [12] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 12 549–12 556.
- [13] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6268–6276.
- [14] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "HiFT: Hierarchical Feature Transformer for Aerial Tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 1–10.
- [15] C. Fu, Z. Cao, Y. Li, J. Ye, and C. Feng, "Siamese Anchor Proposal Network for High-Speed Aerial Tracking," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1–7.
- [16] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning Attentions: Residual Attentional Siamese Network for High Performance Online Visual Tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4854–4863.
- [17] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual Object Tracking Using Adaptive Correlation Filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2544–2550.
- [18] C. Fu, J. Ye, J. Xu, Y. He, and F. Lin, "Disruptor-Aware Interval-Based Response Inconsistency for Correlation Filters in Real-Time Aerial Tracking," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2020.
- [19] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese Instance Search for Tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1420–1429.
- [20] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High Performance Visual Tracking with Siamese Region Proposal Network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8971–8980.
- [21] C. Fu, Z. Cao, Y. Li, J. Ye, and C. Feng, "Onboard Real-Time Aerial Tracking With Efficient Siamese Anchor Proposal Network," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2021.
- [22] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7794–7803.
- [23] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual Attention Network for Scene Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3141–3149.
- [24] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 531–11 539.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in *Advances in neural information processing systems (NIPS)*, 2012, pp. 1097–1105.
- [26] M. Mueller, N. Smith, and B. Ghanem, "A Benchmark and Simulator for UAV Tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 445–461.
- [27] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised Deep Tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1308–1317.
- [28] X. Li, C. Ma, B. Wu, Z. He, and M. Yang, "Target-Aware Deep Tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1369–1378.
- [29] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning Dynamic Siamese Network for Visual Object Tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1781–1789.
- [30] L. Zhang and P. N. Suganthan, "Robust Visual Tracking via Co-Trained Kernelized Correlation Filters," *Pattern Recognition*, vol. 69, pp. 82–93, 2017.
- [31] C. Ma, J. Huang, X. Yang, and M. Yang, "Hierarchical Convolutional Features for Visual Tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3074–3082.
- [32] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-End Representation Learning for Correlation Filter Based Tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5000–5008.
- [33] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient Convolution Operators for Tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6931–6939.
- [34] C. Wang, L. Zhang, L. Xie, and J. Yuan, "Kernel Cross-Correlator," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 32, no. 1, 2018, pp. 4179–4186.
- [35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., "Imagenet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European conference on computer vision (ECCV)*, 2014, pp. 740–755.
- [37] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–17, 2019.
- [38] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7464–7473.
- [39] Y. Wu, J. Lim, and M. Yang, "Online Object Tracking: A Benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2411–2418.