Adaptive and Background-Aware Vision Transformer for Real-Time UAV Tracking

Shuiwang Li[1], Yangxiang Yang[1], Dan Zeng[2,*] and Xucheng Wang[1]
[1]College of Information Science and Engineering , Guilin University of Technology, China
[2]Research Institue of Trustworthy Autonomous Systems,
Southern University of Science and Technology, China
lishuiwang0721@163.com, xyyang317@163.com, zengd@sustech.edu.cn, xcwang@glut.edu.cn

Abstract

While discriminative correlation filters (DCF)-based trackers prevail in UAV tracking for their favorable efficiency, lightweight convolutional neural network (CNN)-based trackers using filter pruning have also demonstrated remarkable efficiency and precision. However, the use of pure vision transformer models (ViTs) for UAV tracking remains unexplored, which is a surprising finding given that ViTs have been shown to produce better performance and greater efficiency than CNNs in image classification. In this paper, we propose an efficient ViT-based tracking framework, Aba-ViTrack, for UAV tracking. In our framework, feature learning and template-search coupling are integrated into an efficient one-stream ViT to avoid an extra heavy relation modeling module. The proposed Aba-ViT exploits an adaptive and background-aware token computation method to reduce inference time. This approach adaptively discards tokens based on learned halting probabilities, which a priori are higher for background tokens than target ones. Extensive experiments on six UAV tracking benchmarks demonstrate that the proposed Aba-ViTrack achieves state-of-the-art performance in UAV tracking. Code is available at `https://github.com/xyyang317/Aba-ViTrack`.

1. Introduction

Unmanned aerial vehicles (UAVs) have been employed in various applications, and recently, UAV tracking has gained considerable attention in visual tracking [37, 4, 66, 67]. However, unlike general visual tracking, UAV tracking poses unique challenges. Common issues such as extreme view angles, motion blur, and severe occlusion can degrade tracking precision. Moreover, the limited battery capacity, computing resources, and low power consumption

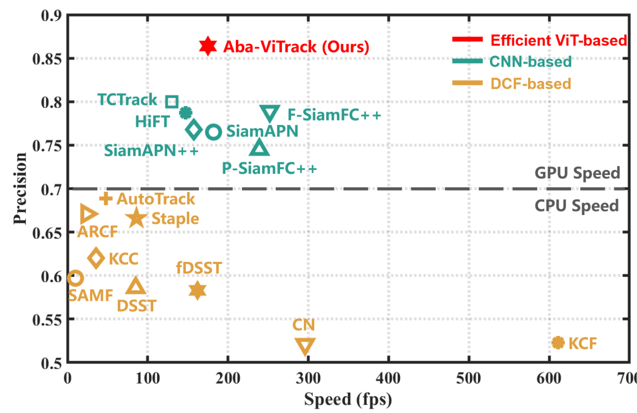

Figure 1. Comparison on UAV123. Compared with DCF-based and CNN-based trackers, our efficient ViT-based tracker (Aba-Track) sets a new record with 0.864 precision and still runs efficiently at around 180 fps.

requirements of UAVs impose stringent demands on efficiency [5, 66, 67, 34]. Therefore, a good UAV tracker must achieve high precision while remaining high efficiency.

As shown in Fig. 1, UAV tracking methods can be broadly divided into two categories: discriminative correlation filters (DCF)-based trackers and deep convolutional neural network (CNN)-based trackers. DCF-based trackers are favored because of their high efficiency derived from operations in the Fourier domain, but they usually achieve low tracking precision [37, 32, 36, 28]. On the other hand, CNN-based trackers can easily obtain high precision, but are not suitable for high-efficiency demands. To combat low efficiency, some lightweight CNN-based trackers are proposed for UAV tracking [5, 66, 67] that employ filter pruning to reduce the parameters of SiamFC++ [70] based on Fisher information [67] or rank information [66, 41], resulting in significant improvements in both precision and efficiency. Very recently, TCTrack [5] has been proposed to utilize temporal contexts to enhance UAV tracking. Different from existing CNN-based tracker, TC-

*Corresponding author.

Track is a hybrid deep learning architecture combining CNN and transformer, where an online temporally adaptive convolution enhances the spatial features with temporal information, and an adaptive temporal transformer refines similarity map. Despite the success in achieving high precision and efficiency, the precision gain is not matched with consideration cost of speed and heavy temporal information use. An even more surprising finding is that exploring visual transformers for UAV tracking remains unexplored.

In this paper, we make the first attempt to utilize efficient Vision Transformers (ViTs) for real-time unmanned aerial vehicle (UAV) tracking. Specifically, we investigate the use of efficient ViTs to enhance the feature learning and template-search coupling processes, thereby making them suitable for real-time UAV tracking. While many lightweight ViTs have been proposed recently through low-rank methods [64], model compression [75, 52, 45], hybrid design [8, 38], they are not well-suited for our purpose for the following reasons. For example, low-rank and quantization-based ViTs often compromise prediction accuracy. Pruning-based ViTs require a time-consuming decision-making process for pruning ratios and subsequent fine-tuning. Hybrid ViTs, which typically employ a CNN-based stem to downsample input images, are unsuitable for our unified framework because the template and search patches have different sizes.

Fortunately, we have efficient Vision Transformers (ViTs) based on conditional computation, such as those proposed in [49] and [73], which can dynamically reduce the number of tokens based on the input. Building on the recent work in A-ViT [73], which proposed an adaptive token reduction mechanism that discards redundant spatial tokens according to dynamical halting probabilities, we present Aba-ViT, an efficient ViT for UAV tracking. Our method incorporates adaptive and background-aware token computation, which learns halting probabilities that are higher for background tokens than target ones through a more informative loss generalized from A-ViT [73]. By taking into account prior knowledge, Aba-ViT is more effective than A-ViT for UAV tracking. This is due to its ability to be aware of the background, which is typically filled with potential distractors and noise that can pose a significant challenge to tracking algorithms. As background tokens are halted with higher probabilities in Aba-ViT without any additional computation burden, our method is expected to reduce the overall compute requirements. As shown in Figure 1, our method sets a new record with a precision of 0.864 and runs efficiently at around 180 frames per second (fps), as compared to DCF- and CNN-based trackers. Extensive experiments on six benchmarks demonstrate that Aba-ViT achieves state-of-the-art performance.

Our contributions can be summarized as follows:

- We make the first attempt to explore using efficient ViTs, particularly in a unified framework, for real-time UAV tracking. The significant improvement in tracking precision with favorable speeds indicates that our effort is very fruitful and worthwhile and may encourage more work in this direction.
- We propose an efficient ViT, Aba-ViT, which incorporates adaptive and background-aware token computation. This allows Aba-ViT to learn halting probabilities that are a priori higher for background tokens than target ones. Using Aba-ViT as the backbone, we have developed a tracker named Aba-ViTrack, which has proven to be an efficient and effective tracker for real-time UAV tracking.
- Our Aba-ViT sets a new state-of-the-art record on six challenging benchmarks, namely UAV123@10fps [48], VisDrone2018 [80], UAVDT [17], UAV123 [48], DTB70 [35], and UAVTrack112_L [20].

2. Related Work

2.1. Visual Tracking

Modern visual trackers can be roughly divided into two classes: DCF-based trackers and DL-based ones. The former prevail in UAV tracking for their more favorable efficiency. Despite their relatively higher efficiency, they hardly maintain robustness under challenging conditions because of the poor representation ability of handcrafted features [34, 37, 28]. To substantially improve tracking precision and robustness, some DL-based trackers have been developed for UAV tracking recently. For instance, Cao et al. [4] proposed a hierarchical feature transformer to achieve interactive fusion of spatial (shallow layers) and semantics cues (deep layers) for UAV tracking. Huang et al. [5] presented a comprehensive framework to fully exploit temporal contexts with a proposed adaptive temporal transformer for aerial tracking. However, the efficiency of these methods is still much lower than most DCF-based trackers. To further improve efficiency of DL-based trackers for UAV tracking, model compression techniques have been recently utilized to reduce model size and thus to improve efficiency [66, 67]. Unfortunately, the model compression methods utilized by these works, though simple and efficient, are still unable to achieve satisfying tracking precision.

Very recently, Cao et al. [5] proposed a framework to exploit temporal contexts for UAV tracking, which significantly outperforms many DCF-based trackers and is apparently superior to the lightweight DL-based trackers just mentioned. However, this method has limitations of inefficient template-search coupling by correlation and multiple modules of relatively independent functions, which has been recognized and addressed with more succinct and unified frameworks recently in generic visual tracking [69, 9, 72, 68]. For example, Xie et al. proposed a Siamese-like dual-branch network in which the features are learned from matching, and ultimately, for matching based solely on Transformers [69]. Cui et al. proposed a Mixed Attention Module (MAM) built upon transformers to unify the process of feature extraction and target information integration [9]. Ye et al. proposed one-stream tracking framework that unifies feature learning and relation modeling and an in-network candidate early elimination module to further improve the inference efficiency [72]. Xie et al. proposed a target-dependent feature network based on the self-/cross-attention scheme, embedding cross-image feature correlation in multiple layers of the feature network so that the output features of the search image can be directly used for predicting target locations without extra correlation step [68]. Although such unified frameworks do bring in efficiency since their more simplified and compact architectures, because of the considerable parameters of the ViTs used, they are still too cumbersome for UAV tracking which places great emphasis on efficiency. In this paper, we explore adapting more efficient ViTs instead for real-time UAV tracking, which, to our knowledge, has not been studied before.

2.2. Efficient Vision Transformers

Transformers, originally designed for NLP [56], have recently demonstrated their great potentials in computer vision [16, 42]. DETR [6] makes the first attempt to apply the transformer model to vision tasks, while ViT [16] first directly apply transformer on non-overlapping image patches for image classification. DeiT [54] further improves the training pipeline with distillation, eliminating the need for large-scale pertaining. And many follow-up works are proposed to refine the architecture [65, 55], explore the relationship between CNN and ViT [10, 25], and build variants of token mixer, e.g., local attention [42], spatial MLP [53], and pooling-mixer [74].

When the inference speed is a major concern, especially on resource-constrained edge devices, efficient ViTs are much desirable. To accelerate ViT, many lightweight ViTs have been proposed recently through low-rank methods [64], model compression [75, 52, 45], hybrid design [8, 38]. However, they do not fit well in with our purpose here. Low-rank and quantization-based ViTs usually sacrifice much accuracy for efficiency. Pruning-based ViTs usually involve the tedious decision of pruning ratios and a finetuning process. Hybrid ViTs with CNN-based stems greatly restrict the input size, namely, images of different input sizes cannot be input simultaneously. With the increased popularity, efficient ViTs based on conditional computation have very recently explored adaptive inference for model acceleration. DynamicViT [49] designs extra control gates to halt tokens, which are trained with the Gumbel-softmax trick, resembling similarities to [57] and [58]. Given Gumbel-softmax-based relaxation solutions might be sub-optimal due to the difficulty of regularization and the heuristic guidance of multi-stage token sparsification, A-ViT [73] exploits an ACT [23]-like approach to remove the need for the extra halting sub-networks, showing improvements on efficiency, accuracy, and token-importance allocation simultaneously. However, A-ViT a priori treats each token equally, i.e., the ponder loss of each token is considered equally important, which neglects the fact that only those tokens with useful information to the downstream tasks rather than noise and distractor are desired. Since the target and background are known in the tracking phase in our visual tracking scenarios, in this paper, we impose larger weights on those tokens containing background, so that they are halted with larger probabilities. With this prior imposed, we call it background-aware A-ViT, dubbed Aba-ViT, and we show that Aba-ViT improves both efficiency and accuracy for UAV tracking.

3. Method

In this section, we present our end-to-end tracking framework, termed as Aba-ViTrack, based on the proposed Aba-ViT backbone. First, we introduce our Aba-ViT for simultaneous feature learning and template-search coupling. This unified scheme enables feature learning and template-search coupling to interact throughout the process, which not only simplifies the process but also makes it more effective, as feature learning becomes more specific while template-search coupling is performed more extensively to better capture the correlation. In addition, the scheme of adaptive and background-aware halting of tokens speeds up model inference. Then, we present the whole framework for UAV tracking, which only includes an Aba-ViT-based backbone and a localization head. An overview of the model is shown in Fig. 2.

3.1. Aba-ViT

Adaptive and background-ware ViT (Aba-ViT) is the key to our seeking of a compact and efficient end-to-end tracker for real-time UAV tracking. The input to Aba-ViT is the target template Z and search image X. They are first split and flattened into sequences of patches, which are then tokenized by a trainable linear projection layer. This process is called patch embedding and results in K tokens, formulated by

$$t^0_{1:K} = \mathcal{E}(Z, X) \in \mathbb{R}^{K\times E}, \tag{1}$$

where E denotes the embedding dimension of each token. Let $\mathfrak{T}^l$ denote the transformer block at layer l, which transforms all tokens from layer $(l-1)$ via $t^l_{1:K} = \mathfrak{T}^l(t^{l-1}_{1:K})$.

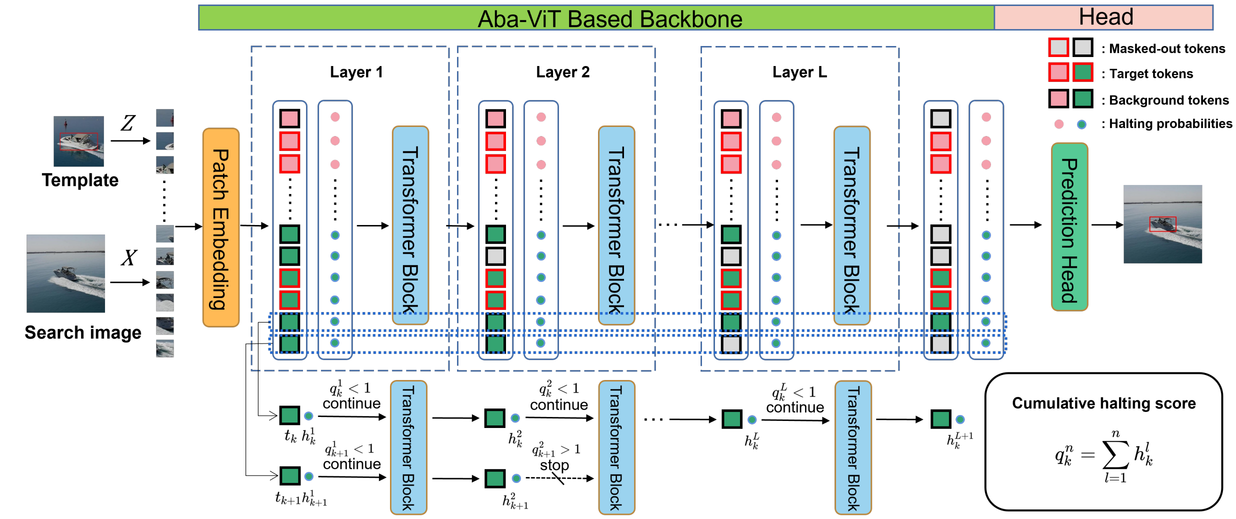

Figure 2. Overview of our framework. It is composed of a single Aba-ViT backbone used for feature learning and template-search coupling and a localization head.

Then the Aba-ViT, denoted by $\mathfrak{A}$, can be formulated by

$$Y = \mathfrak{A}(Z, X) = \mathfrak{T}^L \circ \mathfrak{T}^{L-1} \circ ... \circ \mathfrak{T}^1 \circ \mathcal{E}(Z, X), \quad (2)$$

where $\circ$ denotes the composition operation. The core idea of Aba-ViT is that the tokens can be stopped at earlier layers according to a background-aware halting mechanism, which is dependent on the input. Like A-ViT [73], the halting score h_k^l of the token k at layer l is defined by

$$h_k^l = H(t_k^l) = \sigma(\gamma \cdot t_{k,e}^l + \beta), \quad (3)$$

where $H(\cdot)$ is a halting module implemented by allocating a single neuron into the MLP layer of the existing vision transformer block, $\sigma(u) = \frac{1}{1+e^{-u}}$ is the logistic sigmoid function, $t_{k,e}^l$ indicates the e^{th} dimension of t_k^l, β and γ are shifting and scaling parameters shared across all layers for all tokens. Empirically, the simple choice of $e = 0$ (the first dimension) performs well. As in A-ViT, we stop the token t_k at layer n when its cumulative halting score $q_k^n = \sum_{l=1}^n h_k^l$ exceeds $1 - \epsilon$, i.e., $q_k^n \geqslant 1 - \varepsilon$, where ε is a small positive constant that allows halting after one layer. Once a token is halted, it is masked out by zeroing out its token value and blocking its attention to other tokens, and no update (by transformer block) is applied to it henceforth. Let N_k be the total number of updates applied to t_k, then

$$N_k = \underset{n \leqslant L}{\text{argmin}} \left\{ \sum_{l=1}^{n} h_k^l \geqslant 1 - \varepsilon \right\}. \quad (4)$$

The remainder [23] of t_k is defined as follows

$$R_k = 1 - \sum_{l=1}^{N_k - 1} h_k^l. \quad (5)$$

Finally, the halting probability is defined by

$$p_k^l = \begin{cases} R_k & \text{if } l = N_k, \\ h_k^l & \text{if } l < N_k. \end{cases} \quad (6)$$

This is a valid probability distribution, since it follows directly from the definition that $0 \leqslant p_k^l \leqslant 1$ and $\sum_{l=1}^{N_k} p_k^l = 1$. If no constraints are imposed on the number of updates of each token, it will tend to 'ponder' as long as possible to avoid making mistakes. To limit the amount of computation the network performs, ACT [23] and A-ViT [73] used the following ponder loss to encourage early stopping:

$$\mathcal{L}_{ponder} = \frac{1}{K} \sum_{k=1}^{K} \rho_k = \frac{1}{K} \sum_{k=1}^{K} (N_k + R_k), \quad (7)$$

where ρ_k denotes the ponder loss of the token t_k. However, this loss treats each token equally with weight $1/K$, which seems blind and ignorant when prior knowledge about the tokens is given or learned. In our scenario, it is known in the training phase that a certain token is related to the target or the background. To make use of this information, we generalize the ponder loss $\mathcal{L}_{ponder}$ as follows,

$$\mathcal{L}^*_{ponder} = \frac{1}{K} \sum_{k=1}^{K} \rho_k (\mathrm{I}_{\{t\}}(t_k) + \omega_b \mathrm{I}_{\{b\}}(t_k)), \quad (8)$$

where $\mathrm{I}_{\{t\}}(\cdot)$ and $\mathrm{I}_{\{b\}}(\cdot)$ are indicator functions defined by

$$\mathrm{I}_{\{t(b)\}}(t_k) = \begin{cases} 1 & \text{if } t_k \text{ is a target (background) token,} \\ 0 & \text{otherwise ,} \end{cases} \quad (9)$$

$\omega_b \geqslant 1$ is a predefined constant used to scale the ponder loss of background tokens. Note that $\mathcal{L}^*_{ponder}$ reduces to

$\mathcal{L}_{ponder}$ when $\omega_b = 1$. Similar to previous work on adaptive computation [23, 19], training can be sensitive to the scale factor of the ponder loss $\mathcal{L}_{ponder}$, A-ViT [73] introduced a distributional prior to construct a Kullback-Leibler (KL) divergence for regularization, so that on average all tokens exit at a target depth. Specifically, the halting score distribution that estimates the halting likelihoods distribute across layers, defined by

$$\hat{\mathcal{H}} := \frac{1}{K}\left[\sum_{k=1}^{K} h_k^1, \sum_{k=1}^{K} h_k^2, ..., \sum_{k=1}^{K} h_k^L\right], \tag{10}$$

is pushed toward a predefined Gaussian prior $\mathcal{H} = \mathcal{N}(\mu, \sigma^2)$ via KL divergence $D_{KL}(\cdot)$, where μ and σ are the expected stopping depth and its standard deviation. The distributional prior regularization term is formulated by

$$\mathcal{L}_{distr} = D_{KL}(\hat{\mathcal{H}}||\mathcal{H}). \tag{11}$$

3.2. Aba-ViTrack for UAV Tracking

Overall Architecture. Based on Aba-ViT, we build the Aba-ViTrack, a compact end-to-end tracking framework for UAV tracking. Compared with other prevailing trackers with separate processes of feature extraction and template-search coupling in UAV tracking, it leads to a more compact and neat tracking pipeline only with a single backbone and tracking head. The overall architecture is illustrated in Fig. 2. The input of Aba-ViTrack is a pair of images, i.e., the template $Z \in \mathbb{R}^{3\times H_z\times W_z}$ and the search image $X \in \mathbb{R}^{3\times H_x\times W_x}$. Suppose they are split into patches of size $P \times P$, then the number of patches of Z and X are $K_z = H_zW_z/P^2$ and $K_x = H_xW_x/P^2$, respectively. These patches are concatenated and then fed into the backbone $\mathfrak{A}$, resulting in totally $K = K_z + K_x$ output tokens, denoted by $t^L_{1:K} = [t^L_{1K_z}; t^L_{K_z+1:K}]$, where token sequences $t^L_{1:K_z}$ and $t^L_{K_z+1:K}$ correspond to the template and the search image respectively. Note that the masked-out tokens due to early stopping are replaced with zero tensors without changing the original order of the tokens.

Prediction Head and Loss. Inspired by the corner detection head in [9, 72], we employ a fully convolutional network-based prediction head $\mathcal{C}$ that consists of several Conv-BN-ReLU layers, to directly estimate the bounding box of the target. The output tokens $t^L_{K_z+1:K}$ corresponding to the search image are first reinterpreted to a 2D spatial feature map and then fed into the prediction head, resulting in a target classification score $\mathbf{p} \in [0, 1]^{H_x/P\times W_x/P}$, a local offset $\mathbf{o} \in [0, 1]^{2\times H_x/P\times W_x/P}$, and a normalized bounding box size $\mathbf{s} \in [0, 1]^{2\times H_x/P\times W_x/P}$. The crude target position is estimated by the highest classification score, i.e., $(x_c, y_c) = \text{argmax}_{(x,y)}\mathbf{p}(x, y)$, and the final target bounding box is estimated by

$$[(x_t, y_t); (w, h)] = [(x_c, y_c) + \mathbf{o}(x_c, y_c); \mathbf{s}(x_c, y_c)]. \tag{12}$$

For the tracking task, we adopt the weighted focal loss [30] for classification, a combination of L_1 loss and GIoU loss [50] for bounding box regression. Finally, the overall loss function is:

$$\begin{aligned}\mathcal{L}_{overall} = \mathcal{L}_{cls} + \lambda_{iou}\mathcal{L}_{iou} + \lambda_{L_1}\mathcal{L}_{L_1} \\ +\alpha_p\mathcal{L}^*_{ponder} + \alpha_d\mathcal{L}_{distr},\end{aligned} \tag{13}$$

where the constants $\lambda_{iou} = 2$ and $\lambda_{L_1} = 5$ are set as in [9, 72], α_d as in [73], α_p is set to 0.0001.

4. Experiments

In this section, our method is comprehensively evaluated on six well-known aerial tracking benchmarks, i.e., UAV123 [48], UAVTrack112_L [20], UAV123@10fps [48] , VisDrone2018 [80], UAVDT [17], and DTB70 [35]. All evaluation experiments are conducted on a PC equipped with i9-10850K processor (3.6GHz), 16GB RAM and an NVIDIA TitanX GPU. 40 existing top trackers are included for a thorough comparison, where their results are obtained by running the official codes with their corresponding hyper-parameters. For a clearer comparison, we divide them into two groups, (i) light-weight trackers [66, 67, 4, 5, 3, 1, 31, 22, 14, 11, 28, 37, 33, 26, 44, 13, 39, 60, 59, 15, 62] and (ii) deep trackers [12, 2, 63, 29, 76, 78, 79, 7, 71, 24, 46, 61].

4.1. Implementation Details

Model. We use the proposed efficient vision transformer Aba-ViT as the backbone. The head is a lightweight FCN, consisting of 4 stacked Conv-BN-ReLU layers for each of three outputs. The sizes of the template and search region are set to 128 × 128 and 256 × 256 respectively.

Training. The training splits of GOT-10k [27], LaSOT [18], COCO [40], and TrackingNet [47] are used for training. Batch size is 32. We train the model with AdamW optimizer [43], set the weight decay to 10^{-4}, the initial learning rate for the backbone to 4×10^{-5}, respectively. The total training epochs are set to 300 with 60k image pairs per epoch and we decrease the learning rate by a factor of 10 after 240 epochs.

Inference. During inference, Hanning window penalty is adopted to utilize positional prior in tracking, following the common practice [77]. Specifically, we simply multiply the classification map P by the Hanning window with the same size, and the box with the highest score after multiplication will be selected as the tracking result.

4.2. Comparison with Light-Weight Trackers

In this subsection, our Aba-ViTrack is compared with 25 existing efficient trackers on the standard aerial tracking benchmarks.

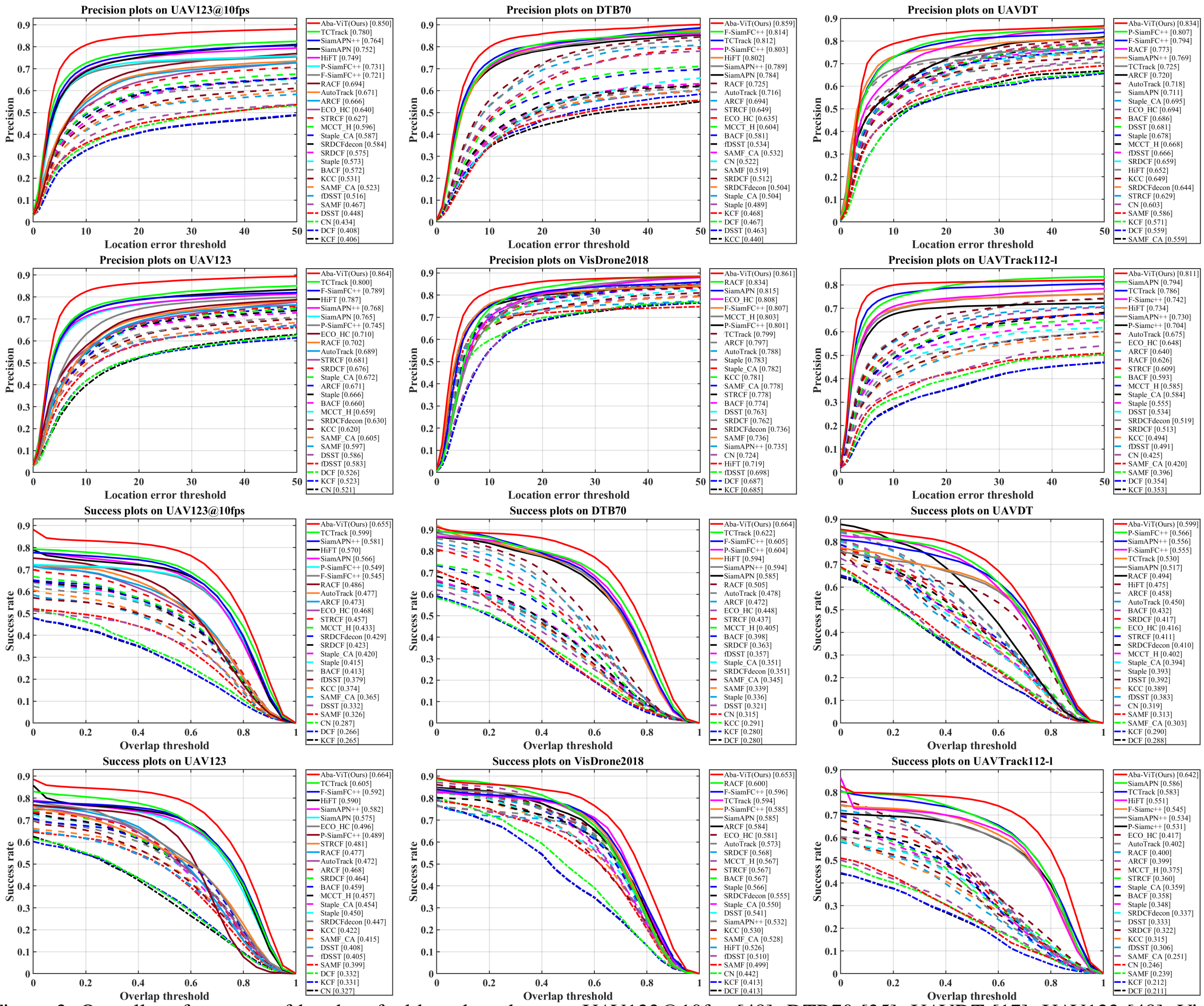

Figure 3. Overall performance of hand-crafted based trackers on UAV123@10fps [48], DTB70 [35], UAVDT [17], UAV123 [48], VisDrone2018 [80], and UAVTrack112_L [20]. Precision and success rate for one-pass evaluation (OPE) are used for evaluation. The precision at 20 pixels and area under curve (AUC) are used for ranking and marked in the precision plots and success plots respectively.

UAV123@10fps: UAV123@10fps [48] is constructed by sampling the UAV123 benchmark from original 30FPS to 10FPS, and is used to study the impact of camera capture speed on tracking performance. **DTB70:** DTB70 [35] consists of 70 UAV sequences, which primarily addresses the problem of severe UAV motion, but also includes various cluttered scenes and objects with different sizes. **UAVDT:** UAVDT [17] is mainly used for vehicle tracking with various weather conditions, flying altitudes and camera views. **UAV123:** UAV123 [48] is a large-scale aerial tracking benchmark involving 123 challenging sequences with more than 112K frames. **VisDrone2018:** VisDrone2018 [80] is from a single object tracking challenge held in conjunction with the European conference on computer vision (ECCV2018), which focuses on evaluating tracking algorithms on drones. **UAVTrack112_L:** UAVTrack112_L [20] is the current biggest long-term aerial tracking benchmark including over 60k frames.

Overall performance evaluation: The overall performance of our Aba-ViTrack with the competing trackers on the six benchmarks is shown in Fig. 3. It can be seen that our Aba-ViTrack outperforms all other trackers on all benchmarks. Specifically, on UAV123@10fps [48] and UAV123 [48], our method significantly outperforms the second-place trackers, respectively, with gains of 7.0% and 6.4% on precision and gains of 5.6%and 5.9% on AUC (i.e., area under curve), respectively. In terms of AUC, our method also surpasses the second-place trackers on DTB70 [35], VisDrone2018 [80], and UAVTrack112_L [20] by 4.2%, 5.3%, and 5.6%, respectively. The least

Table 1. Precision and speed (FPS) comparison between Aba-ViTrack and deep-based trackers on DTB70 [35] . Red, blue and green indicate the first, second and third place.

Tracker	PRC	FPS	Tracker	PRC	FPS
Aba-ViTrack	**85.9**	**185.4**	DiMP18 [2]	79.8	73.0
PrDiMP18 [12]	**84.0**	55.7	DiMP50 [2]	79.2	52.4
PrDiMP50 [12]	76.4	42.1	SiamMask [63]	76.9	**109.6**
SiamRPN++ [29]	79.9	58.2	AutoMatch [76]	82.5	65.2
SiamDW [78]	73.5	65.0	SAOT [79]	83.1	34.0
TransT [7]	**83.6**	53.7	TrSiam [61]	82.7	36.3
SiamGAT [24]	75.1	**92.3**	KeepTrack [46]	**83.6**	19.5
CSWinTT [51]	82.4	9.6	SparseTT [21]	82.3	31.5

precision gain of our method is on UAVTrack112_L [20] over SiamAPN [3] by 1.7%, and the least AUC gain is on UAVDT [17] over P-SiamFC++ [66] by 3.3%. Despite their close performance to ours, these two methods do not always perform well on the other benchmarks. For example, the precision of SiamAPN [3] and P-SiamFC++ [66] on UAV123@10fps [48] is 9.8% and 11.9% lower than ours, respectively. The results show that our method significantly improves the precision and AUC over state-of-the-art methods, and provides a very strong baseline for UAV tracking.

4.3. Comparison with Deep Trackers

The proposed Aba-ViTrack is also compared with fifteen state-of-the-art deep trackers, i.e., DiMP18 [2], PrDiMP18 [12], DiMP50 [2], PrDiMP50 [12], SiamMask [63], SiamRPN++ [29], AutoMatch [76], SiamDW [78], SAOT [79], TransT [7], TrSiam [61], SiamGAT [24], KeepTrack [46], CSWinTT [51], SparseTT [21]. The precision (PRC) and GPU speed of our Aba-ViTrack and the competing deep trackers are shown in Table 1. As can be seen, our Aba-ViTrack achieves the best precision and the fastest GPU speed, suggesting that our method can even beat some deep trackers in both precision and speed. Although several deep trackers' precision is close to ours, such as PrDiMP18, TransT, and KeepTrack, their GPU speeds are much lower. For example, our method is 3 times and 9 times faster than PrDiMP18 and KeepTrack, respectively.

4.4. Evaluation of efficient ViT-based Trackers.

As the study on the effectiveness of leveraging efficient ViTs for UAV tracking is a prime objective in this work, we integrate four different lightweight ViTs, including ViT-tiny [16], DeiT-tiny [54], A-ViT [73], and Aba-ViT, into our ViT-based tracking framework to evaluate their performance for UAV tracking. Their precision (PRC), AUC, and average speed on the six benchmarks are shown in Table 4. We also show eight state-of-the-art trackers' performances in the same table for a thorough comparison, including four DCF-based, i.e., ECO-HC [11], ARCF [28], AutoTrack [37], and RACF [33], and four CNN-based UAV trackers, i.e., HiFT [4], P-SiamFC++ [66] , F-SiamFC++ [67], and TCTrack [5]. As can be seen, the best precision and AUC are basically in the efficient ViT-based class, which can be attributed to the more effective manner of the unified template-search coupling framework and supports the effectiveness of leveraging efficient ViTs for UAV tracking. Among the efficient ViT-based class, Aba-ViTrack achieves the best performance in all six benchmarks except the AUCs on UAV123@10fps [48] and UAVTrack112_L [20] are slightly inferior to DeiT-tiny* by less than 0.5%, and it outperforms the baseline A-ViT* in all benchmarks, justifying the effectiveness of guiding the model to halt background tokens earlier. Note that the speed of Aba-ViTrack is only slightly above A-ViT*, which may be attributed to that they use the same distributional prior on the average token exit length. Better such distributional prior is left to our future work. We also observe that all efficient ViT-based methods achieve real-time GPU and CPU speed. Although their GPU speeds are slower than P-SiamFC++ [66] and F-SiamFC++ [67], their CPU speeds are faster than P-SiamFC++ and close to F-SiamFC++, which may explain the fact that ViT can avoid layer-wise split operation that subjects to convolution (correlation) in CNN, which is short of hardware acceleration in CPU.

Table 2. Ablation study of weighting the ponder loss $\mathcal{L}^*_{ponder}$ on DTB70 [35] with α_p ranging from 0.5×10^{-4} to 1.5×10^{-4}. Note that $\times10^{-4}$ is omitted for simplicity. PRC stands for precision.

α_p	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
PRC	82.9	**85.4**	84.2	83.6	83.4	**85.9**	83.9	**85.1**	82.9	85.1	83.8
AUC	64.6	**65.8**	65.1	65.1	64.6	**66.4**	65.2	**65.7**	64.4	**65.7**	65.5

4.5. Real-World Test

To validate the tracking performance of our method under real-world conditions, we install an embedded onboard processor, the NVIDIA Jetson TX2 4GB, on a typical UAV platform. In real-world UAV testing, the utilization rates of GPU and CPU are 27.7% and 18.9%, respectively, and our tracker remains at an average speed of 35.6 FPS during the tests without the acceleration of TensorRT. We also tested our tracker on a mini PC, specifically an Intel NUC equipped with an i5-1135G7 processor and 16GB RAM, achieving a CPU speed of 43.7 FPS.

4.6. Ablation Study

To verify the effectiveness of our framework, comprehensive ablation studies are presented in this subsection.

Table 3. Ablation study of weighting the background tokens on DTB70 [35] with ω_b ranging from 1.0 to 3.0.

ω_b	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.5	3.0
PRC	84.1	**85.6**	83.1	83.9	83.2	**85.9**	84.1	84.4	**85.5**	82.6	82.5	84.6	84.4
AUC	64.7	**65.9**	64.6	64.7	64.4	**66.4**	64.9	65.3	**65.5**	64.0	64.1	65.3	64.9

Table 4. Evaluation of efficient ViT-based Trackers. Four lightweight ViTs, i.e. ViT-tiny [16], DeiT-tiny [54], A-ViT [73], and Aba-ViT, are integrated into the proposed tracking framework, denoted by ViT-tiny*, DeiT-tiny*, A-ViT*, and Aba-ViT*, respectively. Note that the precision and AUC are shown in form of **(PRC, AUC)**, and the average GPU and CPU speed are shown in form of **[GPU fps, CPU fps]**.

Method		UAV123@10fps [48]	DTB70 [35]	UAVDT [17]	VisDrone2018 [80]	UAV123 [48]	UAVTrack112_L [20]	Avg. FPS [GPU, CPU]
DCF-based	ECO-HC[11]	(64.0, 46.8)	(63.5, 44.8)	(69.4, 41.6)	(80.8, 58.1)	(71.0, 49.6)	(64.8, 41.7)	[— , **83.5**]
	ARCF [28]	(66.6, 47.3)	(69.4, 47.2)	(72.0, 45.8)	(79.7, 58.4)	(67.1, 46.8)	(64.0, 39.9)	[— , 34.2]
	AutoTrack [37]	(67.1, 47.7)	(71.6, 47.8)	(71.8, 45.0)	(78.8, 57.3)	(68.9, 47.2)	(67.5, 40.2)	[— , **57.8**]
	RACF [33]	(69.4, 48.6)	(72.5, 50.5)	(77.3, 49.4)	(83.4, 60.0)	(70.2, 47.7)	(62.6, 40.0)	[— , 35.6]
CNN-based	HiFT [4]	(74.9, 57.0)	(80.2, 59.4)	(65.2, 47.5)	(71.9, 52.6)	(78.7, 59.0)	(73.4, 55.1)	[160.3, —]
	P-SiamFC++[66]	(73.1, 54.9)	(80.3, 60.4)	(**80.7** ,55.6)	(80.1, 58.5)	(74.5, 48.9)	(70.4, 53.1)	[**240.5**, 46.1]
	F-SiamFC++ [67]	(72.1, 54.5)	(81.4, 60.5)	(79.4, 55.5)	(80.7, 59.6)	(78.9, 59.2)	(74.2, 54.5)	[**255.4**, **51.6**]
	TCTrack[5]	(78.0, 59.9)	(81.2, 62.2)	(72.5, 53.0)	(79.9, 59.4)	(80.0, 60.5)	(78.6, 58.3)	[139.6, —]
Efficient ViT-based	ViT-tiny*	(**82.1** , 64.8)	(79.3, 62.4)	(77.0, 55.6)	(83.0, 62.7)	(**83.2**, **65.5**)	(**78.9**, **63.6**)	[166.2, 47.1]
	DeiT-tiny*	(**83.5**, **65.8**)	(**83.6**, **64.9**)	(**81.2**, **58.2**)	(**83.6**, **63.8**)	(82.8, 65.2)	(**80.3**, **64.6**)	[164.6, 46.3]
	A-ViT*	(**82.1**, **65.3**)	(**84.1**, **64.7**)	(78.2, **56.7**)	(**84.4**, **63.9**)	(**82.9**, **66.4**)	(76.8, 62.1)	[176.4, 49.6]
	Aba-ViTrack	(**85.0**, **65.5**)	(**85.9**, **66.4**)	(**83.4**, **59.9**)	(**86.1**, **65.3**)	(**86.4**, **66.4**)	(**81.1**, **64.2**)	[**181.5**, 50.3]

Study on weighting the proposed ponder loss. To see how the weight α_p of the proposed ponder loss $\mathcal{L}^*_{ponder}$ impacts the performance, we train Aba-ViTrack with different α_p that goes from 0.5×10^{-4} to 1.5×10^{-4} in step of 0.1×10^{-4} and evaluate them on DTB70. The precision and AUC are shown in Table 2. As can be seen, the best precision and AUC is at $\alpha_p = 1.0 \times 10^{-4}$. We observe that the maximal difference of precision and AUC is 3.0% and 2.0%, respectively, which suggests that the weight α_p does significantly impact the tracking performance. Appropriately weighted, the proposed ponder loss will lead to better tracking performance, otherwise, it may bring bad effects on the tracking task training.

Study on weighting the background tokens. To understand how the weight ω_b of the ponder loss of background tokens impacts the tracking performance, we train Aba-ViTrack with different ω_b which goes from 1.0 to 3.0 and evaluate them on DTB70 [35]. Note that $\omega_b = 1.0$ reduces to the A-ViT* model. The precision and AUC are shown in Table 3. As can be seen, the best precision and AUC are achieved at $\omega_b = 1.5$. This suggests weighting of the ponder loss of background tokens should be set appropriately, which may be explained by that too large weight may stop too many background tokens so that the discriminative learning lacks sufficient negative samples, thus resulting in degraded performance, whereas, small weight reduces the model to the baseline A-ViT*. If ω_b is appropriately set with fixed α_p, our proposed background-aware ponder loss can improve PRC and AUC of the baseline A-ViT* by 1.8% and 1.7% on DTB70 [35], respectively.

4.7. Qualitative results

Fig. 4 visualizes the token's depth that is adaptively controlled during inference with A-ViT* and our Aba-ViTrack, respectively. The samples are from DTB70 [35], UAV123[48], and UAVTrack112_L [20]. We can observe that our background-aware token halting tends to stop background tokens earlier than A-ViT does, which is, therefore, effective in halting distractors and irrelevant tokens and their associated computations for UAV tracking. For

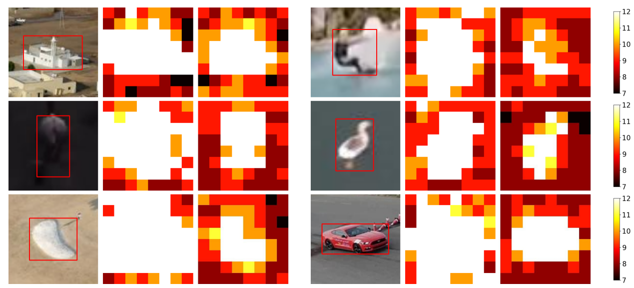

Figure 4. Original image (left), the dynamic token depth of A-ViT (middle), and that of Aba-ViT (right) on samples from the DTB70 [35], UAV123 [48], and UAVTrack112_L [20].

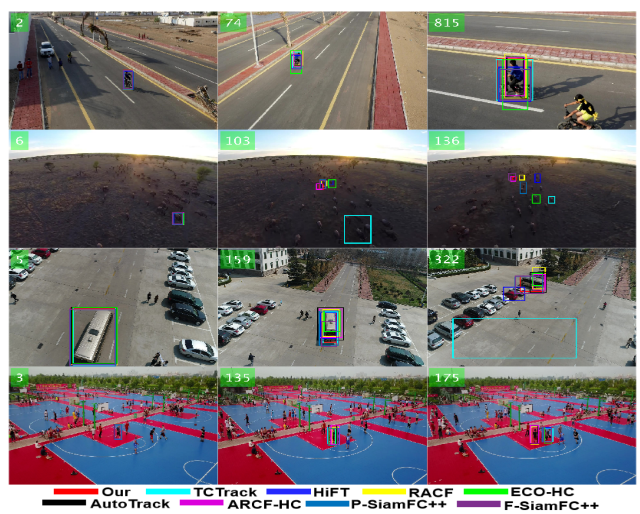

Figure 5. Qualitative evaluation on 4 video sequences from, respectively, UAV123@10fps [48], DTB70 [35], UAVDT [17], and VisDrone2018 [80] (i.e. bike1, Animal1, S1701, and uav000088_0000_s).

example, our approach on animal and person classes basically retains only the target textures, even crude target labels (bounding boxes of targets) are given in the training. The examples of cars and buildings also show similar effects.

Some qualitative tracking results of Aba-ViTrack and eight top trackers are shown in Fig. 5. As can be seen, only our tracker successfully tracks the targets in all challenging examples, where pose variations (i.e., in all sequences), background clusters (i.e., Animal1 and uav000088_0000_s), and scale variations (i.e., bike1 and S1701) are presented. Our method performs much better and is more visually pleasing in these cases, further supporting the effectiveness of the proposed method for UAV tracking.

5. Conclusion

In this work, we make the first attempt to explore using efficient ViTs in a unified template-search coupling framework for real-time UAV tracking. And we proposed a generalized ponder loss to leverage prior information about background and target for background-ware and more effective adaptive halting for UAV tracking. Extensive experiments were conducted to evaluate the effectiveness of the proposed method. Experimental results show that our Aba-ViTrack sets a new state-of-the-art performance on six challenging benchmarks. In the future, we consider extending Aba-ViT to object detection tasks where background and object information is available in the training and study better distributional prior on average token exit length, which may greatly impact the efficiency of our method.

Acknowledgment

Thanks to the support by the Guangxi Science and Technology Base and Talent Special Project (No. GKAD22035127), the National Natural Science Foundation of China (Nos. 62206123, 62066042, 62176170, and 61971005), and the Sichuan Province Key Research and Development Project (No. 2020YJ0282), this work has been made possible.

References

[1] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondřej Mik*vs.*ík, and Philip H. S. Torr. Staple: Complementary learners for real-time tracking. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1401–1409, 2015.

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6182–6191, 2019.

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *European Conference on Computer Vision*, pages 205–221, 2020.

[4] Ziang Cao, Changhong Fu, Junjie Ye, Bowen Li, and Yiming Li. Hift: Hierarchical feature transformer for aerial tracking. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15457–15466, 2021.

[5] Ziang Cao, Ziyuan Huang, Liang Pan, Shiwei Zhang, Ziwei Liu, and Changhong Fu. Tctrack: Temporal contexts for aerial tracking. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14778–14788, 2022.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8122–8131, 2021.

[8] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5260–5269, 2021.

[9] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13608–13618, 2022.

[10] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

[11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6931–6939, 2016.

[12] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7181–7190, 2020.

[13] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference*, 2014.

[14] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *2015 IEEE ICCV*, pages 4310–4318, 2015.

[15] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost van de Weijer. Adaptive color attributes for real-time visual tracking. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, 2014.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

[17] Dawei Du, Yuankai Qi, Hongyang Yu, Yi-Fan Yang, Kaiwen Duan, Guorong Li, W. Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *ECCV,*, pages 375–391, 2018.

[18] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5369–5378, 2018.

[19] Michael Figurnov, Maxwell D. Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry P. Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1790–1799, 2016.

[20] Changhong Fu, Ziang Cao, Yiming Li, Junjie Ye, and Chen Feng. Onboard real-time aerial tracking with efficient siamese anchor proposal network. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99):1–13, 2021.

[21] Zhihong Fu, Zehua Fu, Qingjie Liu, Wenrui Cai, and Yunhong Wang. Sparsett: Visual tracking with sparse transformers. *arXiv preprint arXiv:2205.03776*, 2022.

[22] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1144–1152, 2017.

[23] Alex Graves. Adaptive computation time for recurrent neural networks. *ArXiv*, abs/1603.08983, 2016.

[24] Dongyan Guo, Yan Shao, Ying Cui, Zhenhua Wang, Liyan Zhang, and Chunhua Shen. Graph attention tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9538–9547, 2020.

[25] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. In *International Conference on Learning Representations*, 2021.

[26] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:583–596, 2014.

[27] 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*, (5), 2021.

[28] Ziyuan Huang, Changhong Fu, Yiming Li, Fuling Lin, and Peng Lu. Learning aberrance repressed correlation filters for real-time uav tracking. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2891–2900, 2019.

[29] Minji Kim, Seungkwang Lee, Jungseul Ok, Bohyung Han, and Minsu Cho. Towards sequence-level training for visual tracking. *ArXiv*, abs/2208.05810, 2022.

[30] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision*, 128:642–656, 2018.

[31] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4904–4913, 2018.

[32] Shuiwang Li, Qianbo Jiang, Qijun Zhao, Li Lu, and Ziliang Feng. Asymmetric discriminative correlation filters for visual tracking. *Frontiers of Information Technology & Electronic Engineering*, 21(10):1467–1484, 2020.

[33] Shuiwang Li, Yuting Liu, and et al. Learning residue-aware correlation filters and refining scale estimates with the grabcut for real-time uav tracking. *3DV*, pages 1238–1248, 2021.

[34] Shuiwang Li, Yuting Liu, Qijun Zhao, and Ziliang Feng. Learning residue-aware correlation filters and refining scale for real-time uav tracking. *Pattern Recognition*, 127:108614, 2022.

[35] Siyi Li and D. Y. Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI Conference on Artificial Intelligence*, pages 4140–4146, 2017.

[36] Shuiwang Li, Qijun Zhao, Ziliang Feng, and Li Lu. Equivalence of correlation filter and convolution filter in visual tracking. In *Image and Graphics*, pages 623–634, Cham, 2021. Springer International Publishing.

[37] Yiming Li, Changhong Fu, Fangqiang Ding, Ziyuan Huang, and Geng Lu. Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization, CVPR,11920-11929.

[38] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, S. Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *ArXiv*, abs/2206.01191, 2022.

[39] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision*, pages 254–265, 2014.

[40] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

[41] Mengyuan Liu, Yuelong Wang, and Qiang Sun End to-end representation learning for correlation filter based trackinand Shuiwang Li. Global filter pruning with self-attention for real-time uav tracking. In *British Machine Vision Conference*, 2022.

[42] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[43] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[44] Danelljan M. and et al. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *CVPR*, pages 1430–1438, 2016.

[45] Jiachen Mao, Huanrui Yang, Ang Li, Hai Helen Li, and Yiran Chen. Tprune: Efficient transformer pruning for mobile devices. *ACM Trans. Cyber Phys. Syst.*, 5:26:1–26:22, 2021.

[46] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13424–13434, 2021.

[47] Matthias Mueller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *European Conference on Computer Vision*, 2018.

[48] Matthias Mueller, Neil G. Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *European Conference on Computer Vision*, 2016.

[49] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Neural Information Processing Systems*, 2021.

[50] Seyed Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, 2019.

[51] Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8781–8790, 2022.

[52] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12165–12174, 2022.

[53] Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *Neural Information Processing Systems*, 2021.

[54] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[55] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[57] Andreas Veit and Serge J. Belongie. Convolutional networks with adaptive inference graphs. *International Journal of Computer Vision*, 128:730–741, 2017.

[58] Thomas Verelst and Tinne Tuytelaars. Dynamic convolutions: Exploiting spatial sparsity for faster inference. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2317–2326, 2019.

[59] Chen Wang, Le Zhang, Lihua Xie, and Junsong Yuan. Kernel cross-correlator. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

[60] Ning Wang, Wen gang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4844–4853, 2018.

[61] Ning Wang, Wen gang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1571–1580, 2021.

[62] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *ArXiv*, abs/1704.04057, 2017.

[63] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H. S. Torr. Fast online object tracking and segmentation: A unifying approach. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1328–1338, 2018.

[64] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768, 2020.

[65] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[66] Xucheng Wang, Dan Zeng, Qijun Zhao, and Shuiwang Li. Rank-based filter pruning for real-time uav tracking. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 01–06, 2022.

[67] Wanying Wu, Pengzhi Zhong, and Shuiwang Li. Fisher pruning for real-time uav tracking. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2022.

[68] Fei Xie, Chunyu Wang, Guangting Wang, Yue Cao, Wankou Yang, and Wenjun Zeng. Correlation-aware deep tracking. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8741–8750, 2022.

[69] Fei Xie, Chunyu Wang, Guangting Wang, Wankou Yang, and Wenjun Zeng. Learning tracking representations via dual-branch fully transformer networks. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2688–2697, 2021.

[70] Yinda Xu, Zeyu Wang, Zuoxin Li, Yuan Ye, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI,2020,pp.12549–12556*.

[71] B. Yan, Houwen Peng, Kan Wu, Dong Wang, Jianlong Fu, and Huchuan Lu. Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15175–15184, 2021.

[72] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel,*

October 23–27, 2022, Proceedings, Part XXII, pages 341–357. Springer, 2022.

[73] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022.

[74] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10809–10819, 2021.

[75] Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12135–12144, 2022.

[76] Zhipeng Zhang, Yihao Liu, Xiao Wang, Bing Li, and Weiming Hu. Learn to match: Automatic matching network design for visual tracking. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13319–13328, 2021.

[77] Zhipeng Zhang and Houwen Peng. Ocean: Object-aware anchor-free tracking. *ArXiv*, abs/2006.10721, 2020.

[78] Zhipeng Zhang, Houwen Peng, and Qiang Wang. Deeper and wider siamese networks for real-time visual tracking. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4586–4595, 2019.

[79] Zikun Zhou, Wenjie Pei, Xin Li, Hongpeng Wang, Feng Zheng, and Zhenyu He. Saliency-associated object tracking. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9846–9855, 2021.

[80] Pengfei Zhu, Longyin Wen, and et al. Visdrone-sot2018: The vision meets drone single-object tracking challenge results. In *ECCV Workshops*, pages 469–495, 2018.