Learning Target-Aware Vision Transformers for Real-Time UAV Tracking

Shuiwang Li, *Member, IEEE*, Xiangyang Yang, Xucheng Wang, Dan Zeng, Hengzhou Ye, and Qijun Zhao

***Abstract*— In recent years, the field of unmanned aerial vehicle (UAV) tracking has grown rapidly, finding numerous applications across various industries. While the discriminative correlation filters (DCF)-based trackers remain the most efficient and widely used in the UAV tracking, recently lightweight convolutional neural network (CNN)-based trackers using filter pruning have also demonstrated impressive efficiency and precision. However, the performance of these lightweight CNN-based trackers is still far from satisfactory. In the generic visual tracking, emerging vision transformer (ViT)-based trackers have shown great success by using cross-attention instead of correlation operation, enabling more effective capturing of relationships between the target and the search image. But to best of the authors' knowledge, the UAV tracking community has not yet well explored the potential of ViTs for more effective and efficient template–search coupling for UAV tracking. In this article, we propose an efficient ViT-based tracking framework for real-time UAV tracking. Our framework integrates feature learning and template–search coupling into an efficient one-stream ViT to avoid an extra heavy relation modeling module. However, we observe that it tends to weaken the target information through transformer blocks due to the significantly more background tokens. To address this problem, we propose to maximize the mutual information (MI) between the template image and its feature representation produced by the ViT. The proposed method is dubbed TATrack. In addition, to further enhance efficiency, we introduce a novel MI maximization-based knowledge distillation, which strikes a better trade-off between accuracy and efficiency. Exhaustive experiments on five benchmarks show that the proposed tracker achieves state-of-the-art performance in UAV tracking. Code is released at: https://github.com/xyyang317/TATrack.**

***Index Terms*— Real time, target aware, unmanned aerial vehicle (UAV) tracking, vision transform.**

Manuscript received 27 December 2023; revised 10 May 2024 and 2 June 2024; accepted 15 June 2024. Date of publication 21 June 2024; date of current version 3 July 2024. This work was supported in part by Guangxi Natural Science Foundation under Grant 2024GXNSFAA010484; in part by Guangxi Science and Technology Base and Talent Special Project under Grant GKAD22035127; in part by Guangxi Key Technologies Research and Development Program under Grant AB2304900; in part by the National Natural Science Foundation of China under Grant 62206123, Grant 62066042, Grant 62262011, Grant 62176170, and Grant 61971005; and in part by Sichuan Province Key Research and Development Project under Grant 2020YJ0282. *(Corresponding authors: Dan Zeng; Qijun Zhao.)*

Shuiwang Li, Xiangyang Yang, Xucheng Wang, and Hengzhou Ye are with Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology, Guilin 541004, China (e-mail: lishuiwang0721@163.com; xyyang317@163.com; xcwang@glut.edu.cn; yehengzhou@glut.edu.cn).

Dan Zeng is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: zengd@sustech.edu.cn).

Qijun Zhao is with the College of Computer Science, Sichuan University, Chengdu 610017, China (e-mail: qjzhao@scu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2024.3417400

I. Introduction

VISUAL tracking is an essential and demanding endeavor in the realm of computer vision and pattern recognition, teeming with extensive applications spanning a myriad of fields [1], [2], [3], [4], [5], [6]. Especially, armed with a specialized visual tracking algorithm, unmanned aerial vehicle (UAV), which has recently witnessed a surge in popularity across a diverse array of applications, can be used for target following, surveillance, aerial cinematography, aircraft refueling, search and rescue operations, precision agriculture, and among others [7], [8], [9], [10]. UAV-based object tracking, which is called UAV tracking, attempts to infer and anticipate the location and scale of an arbitrary object in subsequent aerial image frames given an initial state in the first aerial image frame [7], [11], [12]. Real-time UAV tracking refers to a technological feature that enables the accurate, continuous, and live monitoring and tracking of targets at real-time speed [not less than 30 frames/s (FPS)] with limited computing resources onboard a UAV. Although general visual tracking algorithms can be adapted to UAV tracking directly, UAV tracking presents several distinctive challenges not typically found in general visual tracking scenarios. These challenges encompass extreme viewing angles, motion blur, and significant occlusion, all of which can lead to a decline in tracking algorithms' precision. Furthermore, UAV tracking imposes rigorous efficiency requirements due to factors, such as limited battery capacity, constrained computing resources, and low power consumption needs for UAVs [2], [4], [13], [14]. As a result, it is crucial to design UAV tracking algorithms that successfully strike a balance between efficiency and accuracy, ensuring optimal operation in demanding conditions.

At present, UAV tracking methods may be roughly categorized into two distinct types: those based on discriminative correlation filters (DCFs) and deep convolutional neural networks (CNNs). The choice between these two methodologies often requires balancing the demand for high efficiency against the need for high precision. DCF-based trackers have been favored due to their efficient operations in the Fourier domain [7], [11], [15]. However, they often struggle to achieve high tracking precision. On the other hand, CNN-based trackers are well known for their ability to achieve high precision, but they frequently necessitate substantial computational resources. This requirement makes them less amenable to situations demanding high efficiency. To address this trade-off, researchers have introduced lightweight CNN-based trackers

for UAV tracking. These trackers employ filter pruning techniques [2], [9], [10], [16] to reduce the number of parameters in the network, leading to significant improvements in both precision and efficiency. Very recently, TCTrack [2], a hybrid deep learning architecture that combines CNN and transformer models, has been proposed to enhance UAV tracking. The architecture uses an online temporally adaptive convolution to enhance the spatial features with temporal information, and an adaptive temporal transformer to refine the similarity map. While TCTrack has demonstrated remarkable precision and efficiency, it involves specific trade-offs. The precision gain comes at the cost of increased computational demand and extensive utilization of temporal information, which has an adverse effect on the tracker's overall speed. In addition, the increased complexity of the model demands a larger quantity of training data and lengthier training times. It is more important to note that the adaption of transformer architecture in TCTrack is concentrated primarily on refining the similarity map rather than honing the template–search coupling. Nonetheless, recent advancements have shown that template–search coupling with vision transformers (ViTs) can be notably successful in generic visual tracking [3], [17], [18], [19], [20], [21]. We believe that it is worthwhile to explore the potential of ViTs to deliver more effective and efficient template–search coupling for UAV tracking. But to best of the authors' knowledge, the potential of ViTs to deliver more effective and efficient template–search coupling for UAV tracking remains largely unexplored in the UAV tracking community. One possible reason for this is that studies focused on general visual tracking did not pay enough attention to efficiency, leading to a large number of intimidating model parameters and unsatisfactory running speed. This, in turn, has deterred many beneficial explorations in this area. Therefore, developing efficient ViT-based trackers for real-time UAV tracking is still an interesting and challenging problem that needs to be explored.

In this article, we dedicate our effort to developing efficient trackers with ViTs for real-time UAV tracking. In our framework, feature learning and template–search coupling are integrated into an efficient one-stream ViT to avoid inefficient template–search coupling by correlation or heavy relation modeling modules. We demonstrate that it is feasible to create ViT-based trackers for UAV tracking while striking a favorable balance between precision and efficiency. Surprisingly, we show that the ViT-based trackers we propose can all operate at real-time speeds with only a single CPU. In addition, we bring to light a previously unnoticed issue when performing template–search coupling with ViTs. Specifically, the template and search image patches have unequal sizes, which leads to an abundance of background tokens relative to target tokens. Given that transformer blocks are intended to represent target/background tokens using all input tokens, the target information are prone to being diminished through transformer blocks due to its minority status. To address this problem, we propose maximizing the mutual information (MI) between the template image and the corresponding feature representation generated by the ViT, so that crucial target information are preserved in this process. At its core, the MI we measure reveals the dependency between the input template and its feature representation, by maximizing which the feature representation of the template is expected to maintain the strongest dependency on the target template. We refer to this adapted ViT as a target-aware ViT, which forms the basis of our proposed target-aware tracker, termed TATrack. It is worth noting that the maximization of MI is solely conducted during the training process, ensuring no additional computational burden is added during the inference phase. To further enhance efficiency, we introduce a novel knowledge distillation method based on maximizing MI. This approach compresses the tracker, striking a better balance between accuracy and efficiency. By maximizing MI between the feature representations of the teacher and student models, we ensure that the student model captures the most pertinent information from the teacher model's representation. This yields better generalization and performance, especially in the presence of noise, as MI is less sensitive to noise and outliers compared with the widely used mean squared error (MSE). A selection of five efficient ViTs are chosen for feature extraction and template–search coupling to validate the proposed method. Extensive experiments on a comprehensive range of five benchmarks provide solid evidence that our method is able to deliver cutting-edge performance. As shown in Fig. 1, while the precision of CNN-based trackers is basically above that of DCF-based ones, our ViT-based trackers surpass CNN- and DCF-based trackers consistently in precision. Moreover, our methods also outperform several DCF- and CNN-based ones in terms of running speeds. Note that the speed of DCF-based trackers is evaluated on a CPU while that of CNN- and ViT-based ones is evaluated on a graphics processing unit (GPU). And, our TATrack-DeiT sets a new record with a precision of 84.9 and runs efficiently at around 242.3 FPS. Our contributions can be summarized as follows.

1) We propose to develop real-time UAV trackers based on efficient ViTs, particularly in a unified framework. The substantial improvement in tracking precision at favorable speeds highlights the fruitfulness and significance of our effort. We anticipate that our work will inspire future advancements in this direction.
2) We propose to learn target-aware ViTs by maximizing the MI between the template image and its feature representation for UAV tracking. The proposed tracker, named TATrack, has proven to be an efficient and effective tracker for real-time UAV tracking.
3) We propose a novel MI maximization-based knowledge distillation to further enhance efficiency, with empirical evidence showing a significant increase in tracking speed while only minimally reducing accuracy.
4) Our TATrack sets a new state-of-the-art record on five challenging benchmarks, namely, DTB70 [22], UAVDT [23], VisDrone2018 [24], UAV123 [25], and UAV123@10 FPS [25].

The rest of this article is organized as follows. Section II presents an overview of the previous research related to this study. Section III revisits self-attention in the ViT. In Section IV, we detail the methodology of the proposed

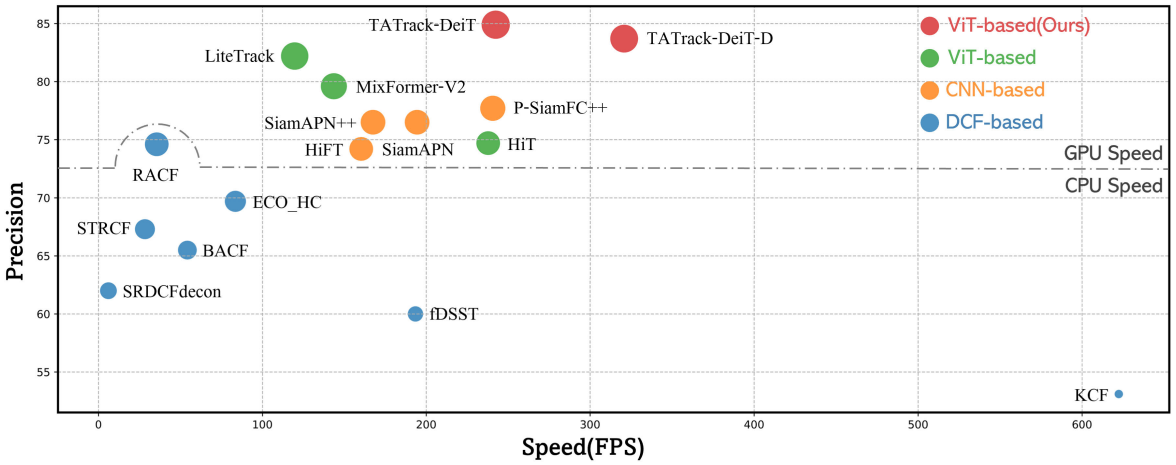


Fig. 1. Compared with DCF-based, CNN-based, and ViT-based lightweight trackers, our ViT-based trackers surpass CNN- and DCF-based trackers consistently in precision, and our tracker TATrack-DeiT sets a new record with 84.9 precision and still runs efficiently at around 242.3 FPS. Note that the speed of DCF-based trackers is evaluated on a single CPU while those of CNN- and ViT-based trackers are evaluated on a single GPU.

technique. Section V outlines the conducted experiments and discusses their results. Finally, Section VI provides the conclusion and insights drawn from this article.

II. Related Works

This section offers a brief survey on visual tracking methods and ViTs, including DCF- and deep learning (DL)-based tracking approaches and the application of ViTs to visual tracking, based upon which the motivations of our work are highlighted.

A. Visual Tracking

When it comes to modern visual trackers, there are two main classes: DCF-based trackers and DL-based trackers. Within the context of UAV tracking, DCF-based trackers are preferred for their high efficiency, since fast Fourier transform (FFT) allows correlation to be evaluated in the frequency domain, and the handcrafted features they often use are computationally very effective [2], [4], [7], [9], [10], [15]. However, despite their commendable efficiency, DCF-based trackers often struggle to maintain robustness under challenging conditions due to the limited representation capacity of handcrafted features they rely on [4], [7], and [11]. Many recent studies have introduced DL-based trackers to enhance tracking precision and robustness in UAV tracking [2], [8], [26]. However, they often lack efficiency compared with DCF-based trackers. To address this, researchers have explored model compression techniques to reduce DL-based model sizes and enhance efficiency [9], [10]. While promising, these techniques struggle to maintain satisfactory tracking precision. In addition, these DL-based trackers for UAV tracking face challenges with ineffective template–search coupling by correlation. Recently, there has been an increasing focus on developing succinct and unified frameworks for generic visual tracking using ViTs. For instance, Xie et al. [17] proposed a Siamese-like dual-branch network that utilizes ViTs to learn features from matching and ultimately match based solely on Transformers. To unify target information integration and feature extraction into a tracking framework, Cui et al. [18] proposed a mixed attention module (MAM) based on Transformers. Ye et al. [3] proposed a one-stream tracking framework that unifies feature learning and relation modeling using ViTs. Although the use of ViTs in a unified tracking framework has demonstrated promise in elevating the accuracy and efficiency of generic visual tracking, their significant parameter sizes pose a challenge for deploying them in real-time tracking applications, particularly in UAV tracking. In this article, we explore adapting more efficient ViTs for real-time UAV tracking, which, to our knowledge, has not been well studied before.

It is worthy noting that there are several works closely related to our study, which also aim at enhancing target awareness of trackers for visual tracking. For example, Li et al. [27] proposed a novel scheme to learn target awareness by developing a regression loss and a ranking loss to guide the generation of target-active and scale-sensitive features. This framework is based on a Siamese matching network and determines the significance of each convolutional filter and chooses target-aware features based on activations to represent the targets. Guo et al. [28] proposed to learn target-aware representation for visual tracking via informative interactions. The Siamese-like backbone networks (InBNs), whose central component is a general interaction modeler (GIM) that injects the target information into various stages of the backbone network, were used to target awareness by executing numerous branchwise interactions inside them. Despite the same end of target awareness, our approach is quite different from these methods. On the one hand, in our method template and search images are coupled in a unified framework rather than a Siamese-like manner; on the other hand, we enforce target awareness by maximizing the MI between the input template and its feature representation, which is quite different from the methods just mentioned.

B. Vision Transformers

Transformers, initially designed for natural language processing (NLP) [29], have recently shown great promise in

computer vision tasks [30], [31]. The first attempt to apply transformer models to vision tasks was made by detection transformer (DETR) [32], which demonstrated its effectiveness in object detection. ViT [30] was the first to directly apply transformers on nonoverlapping image patches for image classification, achieving competitive results with traditional CNNs. DeiT [33] improved the training pipeline of ViT by introducing distillation, which eliminates the need for large-scale pretraining. Many follow-up works have been proposed to refine the architecture of ViT and build variants of token mixers [34], [35], such as local attention [31], spatial MLP [36], and pooling mixer [37], aiming to further improve its performance. The recent application of ViT in learning the connections among pixels within distinct small segments of an image has been employed to calculate the correlation between template and search images in visual tracking [3], [17], [18], [19], showcasing remarkable results. More importantly, these transformer-based trackers combined feature learning with template–search coupling into a unified framework in visual tracking, breeding a new tracking paradigm. However, these methods failed to recognize the fact that the background tokens significantly outnumber the target ones as the search image is expected to be obviously larger than the template image to deal with large translations of targets between neighboring frames. This disproportion could diminish the target information as the background information prevails in the basis for representing tokens via transformer blocks.

To tackle this issue and strive for efficient ViT-based trackers in UAV tracking, in this work, we propose to maximize the MI between the template image and its feature representation generated by the ViT. Learning a representation that maximizes/minimizes the MI between the input and its representation, possibly subject to some structural constraints, where MI is a measure of the mutual dependence between the two random variables, is the so-called InfoMax principle [38]. In this article, we use MI to quantify the "amount of information" obtained about the feature representation of the target with ViT by observing the template image. To maximize MI, our goal is to retain as much target information as possible in the feature representation produced by ViTs. This kind of ViT, which aims to keep the target information intact, is referred to as target aware. By maintaining target awareness, we strive to enhance the tracking precision of ViTs in UAV tracking applications. Since this procedure takes place solely during the training phase, it does not impose any extra computational load during the inference stage. This is well-suited for UAV tracking, where efficiency is of paramount importance.

C. Knowledge Distillation

Knowledge distillation is a technique that compresses models, transferring knowledge from a complex "teacher" model to a simpler "student" model [39], [40]. Its main aim is to distill the teacher model's knowledge into a more compact form, making it suitable for resource-constrained environments [41]. Leveraging the teacher model's knowledge enables the student model to achieve comparable or superior performance with reduced computational resources and memory usage [42]. In knowledge distillation, knowledge types, distillation strategies, and the teacher–student architectures play a crucial role in student learning [40]. Knowledge distillation typically falls into three categories based on the knowledge type: response-based, feature-based, and relation-based distillation [40], [41], [43]. It has been widely used in various machine learning tasks, including image classification [44], object detection [45], [46], and neural machine translation [47], to improve the efficiency and effectiveness of deep learning models. Recently, various techniques in knowledge distillation are exploited to transfer knowledge from teacher models to student models effectively, ultimately improving the efficiency of DL-based trackers for visual tracking. For example, Li et al. [48] presented a mask-guided self-distillation to compress the models of Siamese-based visual trackers, which enables Siamese-based visual trackers to capture crucial knowledge for effecting the performance of tracking. Sun et al. [49] presented a lightweight dual Siamese network for onboard hyperspectral object tracking in which a joint spatial–spectral knowledge distillation method is designed to teach a lightweight dual Siamese tracker to learn from a deep tracker. Zhao et al. [50] introduced a distillation-ensemble-selection framework, where several student trackers are crafted through knowledge distillation from a designated teacher tracking model. An ensemble module amalgamates the outputs of these student trackers using a learnable fine-grained attention module. During the online tracking phase, a selection module dynamically manages the tracker's complexity by pinpointing a suitable subset of candidate tracker models. Although these techniques have proven effective in improving the efficiency of DL-based trackers for visual tracking, they are primarily Siamese-based and customized for specific tracking frameworks and architectures. Adapting them to our ViT-based approach is not straightforward due to differences in methodology and architecture. In this work, we introduce a simple yet effective knowledge distillation method based on maximizing MI. To ensure that the student model captures the most pertinent information from the teacher model's representation, we propose maximizing MI between the feature representations of the teacher and student models. This approach leads to improved generalization and performance, particularly in noisy environments, as MI is less affected by noise and outliers compared with the commonly used MSE.

III. Revisit Self-Attention in ViT

It is well-established that ViTs can outperform CNNs, such as ResNets, in image recognition [51]. The robustness and superiority of ViT features can be primarily attributed to the flexible and dynamic receptive fields that are made possible by the self-attention mechanism. Self-attention operates by calculating a weighted average of feature representations. The weights are determined based on the similarity score between pairs of representations. This allows the model to assign higher weights to more relevant features, resulting in a more accurate representation of the data. By incorporating self-attention, ViT models can effectively capture long-range dependencies and establish meaningful connections between different parts of the input data. This flexibility and dynamic nature of the receptive

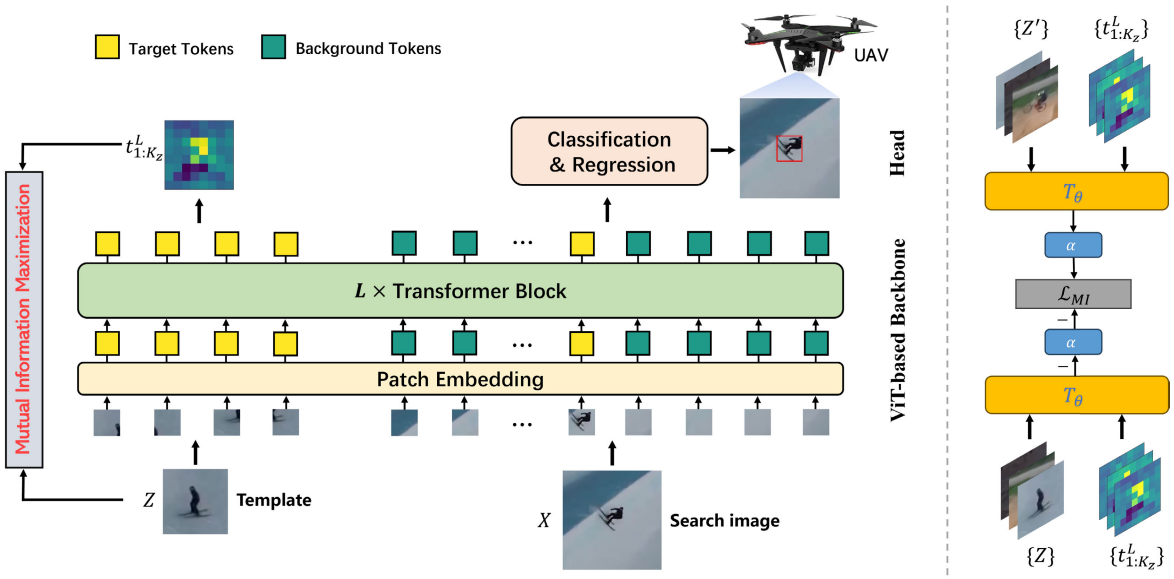


Fig. 2. (Left) Overview of our framework. It is composed of a single efficient ViT-based backbone used for feature learning and template–search coupling and a localization head. (Right) Details of the MI maximization module. Note that $\{Z\}$ indicates a batch of Z, and $\{Z'\}$ denotes the randomly shuffled of $\{Z\}$.

fields contribute to the exceptional performance of ViT models in various computer vision tasks. Nevertheless, caution should be exercised when applying ViT to visual tracking as we will discuss in the following.

Formally, an input sequence of n tokens of dimensions d, $\mathcal{X} \in \mathbb{R}^{d\times n}$, is first projected using three matrices $W_Q \in \mathbb{R}^{d_q\times d}$, $W_K \in \mathbb{R}^{d_k\times d}$, and $W_V \in \mathbb{R}^{d_v\times d}$, with $d_k = d_q$, to extract feature representations Q, K, and V, respectively, which are referred to as query, key, and value correspondingly. More specific, $Q = W_Q\mathcal{X} = \{q_1, \ldots, q_n\}$, $K = W_K\mathcal{X} = \{k_1, \ldots, k_n\}$,$V = W_V\mathcal{X} = \{v_1, \ldots, v_n\}$. Finally, self-attention can be written as follows:

$$\begin{aligned}\mathcal{Y} &= \textbf{Attention}(Q, K, V) = \textbf{Softmax}\left(K^T Q/\sqrt{d_q}\right)V \\ &= \{y_1, \ldots, y_n\}\end{aligned} \tag{1}$$

where **Softmax**$(\cdot)$ denotes a rowwise softmax normalization function and $\mathcal{Y}$ is the representation of $\mathcal{X}$ after applying the self-attention. Thus, each token of $\mathcal{Y}$ depends on all tokens of $\mathcal{X}$. To be more specific, for a given token x_i, its representation after a self-attention layer is

$$\begin{aligned}y_i &= \textbf{Attention}(q_i, K, V) \\ &= \sum_{l=1}^{n} \textbf{Softmax}\left(k_l^T q_i/\sqrt{d_q}\right)v_l = \sum_{l=1}^{n} s_{i,l}v_l\end{aligned} \tag{2}$$

where $s_{i,l} := \textbf{Softmax}(k_l^T q_i/(d_q)^{1/2}) \geq 0$ is considered a similarity score between token x_i and x_l. Let $x_i \sim x_j$ be that token x_i and x_j are semantically related. Denoted by $I_R(i)$ the index set that indexes the tokens semantically related to x_i, $\mathrm{I}_{R^c}(i)$ its complementary set, i.e., $\mathrm{I}_R(i) \cup \mathrm{I}_{R^c}(i) = \{1, \ldots, n\}$. As y_i is a convex combination of $\{v_l\}_{l=1}^n$, $\sum_{l\in \mathrm{I}_{R^c}(i)} s_{i,l}$ is expected to vanish or be very small; otherwise, y_i is partially represented by distractive or noise information for $\sum_{l=1}^n s_{i,l}v_l$ is semantically unrelated to x_i. However, $\sum_{l\in \mathrm{I}_{R^c}(i)} s_{i,l}$ increases as $|\mathrm{I}_{R^c}(i)|$ [i.e., the cardinality of the $\mathrm{I}_{R^c}(i)$] grows, because $s_{i,l} > 0$ is always true in practice. In the specific visual tracking scenario concerned here, when x_i represents a target token, the set $|\mathrm{I}_{R^c}(i)|$ is notably larger than $|\mathrm{I}_R(i)|$. Typically, $|\mathrm{I}_{R^c}(i)|$ is more than four times larger than $|\mathrm{I}_R(i)|$ to account for significant target translations. As a result, the information in y_i that is not semantically related to x_i can occupy a significant portion, which can potentially weaken the representation of the target information. To combat the loss of target information, in this work, we propose to maximize the MI between the template image and its feature representation produced by ViT, which will be detailed in Section IV.

IV. Methodology

An overview of the proposed method TATrack is illustrated in Fig. 2. Our framework consists of a target-aware ViT-based backbone, denoted by TA-ViT, and a prediction head. The backbone carries out feature learning and template–search coupling concurrently, allowing both processes to interact throughout the procedure. This not only streamlines the process but also enhances its effectiveness, as feature learning becomes more specialized while template–search coupling is executed more comprehensively to better capture the correlation. The input to TATrack includes a target template Z and a 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 $\mathcal{K}$ tokens, which are formulated by

$$t^0_{1:\mathcal{K}} = \mathcal{E}(Z, X) \in \mathbb{R}^{\mathcal{K}\times d} \tag{3}$$

where d denotes the embedding dimension of each token, token sequences $t^0_{1:\mathcal{K}_z}$ and $t^0_{\mathcal{K}_z+1:\mathcal{K}}$ correspond to the template and search image, respectively, such that $\mathcal{K} = \mathcal{K}_z + \mathcal{K}_x$. Let $\mathcal{T}^l$ be the transformer block at layer l, which transforms all tokens from layer $(l-1)$ via $t^l_{1:\mathcal{K}} = \mathcal{T}^l(t^{l-1}_{1:\mathcal{K}})$. Then, the backbone TA-ViT, denoted by $\mathfrak{B}$, can be formulated by

$$Y = \mathfrak{B}(Z, X; \varphi) = \mathcal{T}^L \circ \mathcal{T}^{L-1} \circ, \ldots, \circ \mathcal{T}^1 \circ \mathcal{E}(Z, X; \varphi) \tag{4}$$

where $\circ$ denotes the composition operation and φ parameterizes $\mathfrak{B}$. The core idea of TA-ViT is that the MI between the template image and its feature representation is maximized, which will be detailed in Section IV-A.

A. Learn Target-Aware ViTs With MI Maximization

In order to provide context for achieving target-aware ViTs with MI maximization, it is important to first understand some key concepts related to this approach. MI maximization is a technique used in unsupervised learning to measure the amount of information that is shared between two random variables. Let $x \in \mathfrak{X}$ and $y \in \mathfrak{Y}$ be two random variables. The MI between x and y, denoted by $I(x, y)$, formally can be expressed as follows:

$$\begin{aligned} I(x, y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(x, y)}{p(x)p(y)}\right] \\ &= D_{\mathrm{KL}}(p(x, y)||p(x)p(y)) \end{aligned} \tag{5}$$

where $p(x, y)$ denotes the joint probability distribution, $p(x)$ and $p(y)$ are the marginals, and D_{KL} is the Kullback–Leibler divergence (KLD). In practice, estimating MI can be challenging since we typically have access to samples but not the underlying distributions [52]. In this work, we utilize the MI estimator Deep InfoMax [53], which is based on Jensen–Shannon divergence (JSD) instead of KLD, to learn target-aware ViTs for UAV tracking. The choice of JSD as the divergence measure in the Deep InfoMax estimator offers several advantages. JSD is a symmetrized version of the KLD, which makes it more robust and less sensitive to differences in the distributions being compared. This can be beneficial when dealing with samples and limited access to the true underlying distributions. This JS MI estimator, denoted by $\hat{\mathfrak{J}}^{(\mathrm{JSD})}(x, y; \theta)$, is defined as follows:

$$\begin{aligned} &\hat{\mathfrak{J}}^{(\mathrm{JSD})}(x, y; \theta) \\ &\quad = \mathbb{E}_{p(x,y)}\left[-\alpha(-T_\theta(x, y))\right] - \mathbb{E}_{p(x)p(y)}\left[\alpha(T_\theta(x, y))\right] \end{aligned} \tag{6}$$

where $T_\theta : \mathfrak{X} \times \mathfrak{Y} \to \mathbb{R}$ is a neural network parameterized by θ, $\alpha(z) = log(1 + e^z)$ is the softplus function. The right part of Fig. 2 depicts the structure of the Jensen–Shannon MI estimator. Note that $\{Z\}$ indicates a batch of Z, and $\{Z'\}$ denotes the randomly shuffled of $\{Z\}$, ensuring Z is linearly independent of $t^L_{1:\mathcal{K}_z}$ to simulate the marginal distribution between Z and $t^L_{1:\mathcal{K}_z}$. We propose to maximize the Jensen–Shannon MI estimator in the pursuit of target-aware ViTs for UAV tracking. Specifically, we aim to maximize the following objective:

$$\begin{aligned} &\hat{\mathfrak{J}}^{(\mathrm{JSD})}\left(Z, t^L_{1:\mathcal{K}_z}; \theta, \varphi\right) \\ &\qquad = \hat{\mathfrak{J}}^{(\mathrm{JSD})}\left(Z, [\mathfrak{B}(Z, X; \varphi)]_{1:\mathcal{K}_z}; \theta, \varphi\right) \end{aligned} \tag{7}$$

where $t^L_{1:\mathcal{K}_z} = [\mathfrak{B}(Z, X; \varphi)]_{1:\mathcal{K}_z}$ corresponds to the feature representation of the target template Z with the ViT $\mathfrak{B}$. The loss we use for MI maximization is finally defined as follows:

$$\mathcal{L}_{\mathrm{MI}} = -\hat{\mathfrak{J}}^{(\mathrm{JSD})}\left(Z, t^L_{1:\mathcal{K}_z}; \theta, \varphi\right). \tag{8}$$

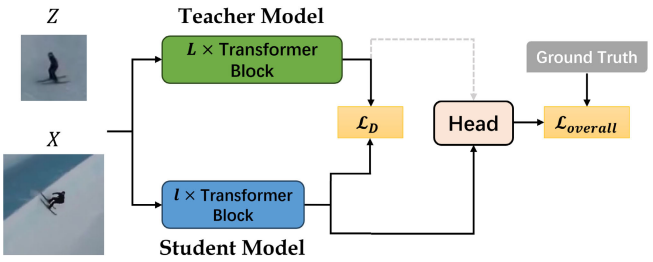


Fig. 3. Framework of the proposed MI maximization-based knowledge distillation.

B. MI Maximization Knowledge-Based Distillation

TATrack aims to become an efficient UAV tracker, striving to enhance efficiency without compromising accuracy too much. To achieve this goal, we propose the MI maximization-based distillation method to make TATrack more efficient, the distillation structure is as shown in Fig. 3. Our method is feature-based, meaning that a smaller model (student) learns from a bigger one (teacher) by focusing on the features the bigger model has learned. Instead of copying the exact predictions, the smaller model tries to match the internal features of the bigger model. This helps the smaller model to learn important features and can improve its performance, especially when resources are limited. Instead of relying on the commonly used MSE to measure the difference between the two feature representations, we propose maximizing the MI between the feature representations of the teacher and student models, which yields the better generalization and performance, especially in the presence of noise, as MI is less sensitive to noise and outliers compared with MSE [39].

In addition, the choice of teacher–student architectures is pivotal in the process of knowledge distillation. These architectures determine how the knowledge learned by the teacher model is transferred to the student model. By selecting appropriate architectures, we can ensure that the student model effectively learns from the teacher's knowledge, leading to improved performance and efficiency. Considering the unlimited choices of student models, we opt for a self-similar architecture to construct the student model, specifically, the student has the same architecture as the teacher but with a smaller ViT as backbone (with fewer ViT blocks), which offers the following advantages. First, this choice simplifies the design process, which facilitates easier implementation and training, as well as better interpretability of the model's behavior. Second, it promotes the modularity and scalability, allowing for easy expansion or modification of the model as needed. Given the teacher–student architecture, we use the Jensen–Shanno MI estimator to implement the MI maximization for knowledge distillation, resulting in the objective function of our MI maximization knowledge-based distillation

$$\mathcal{L}_D = -\hat{\mathfrak{J}}^{(\mathrm{JSD})}\left(t^L_{1:\mathcal{K}_z}, t^l_{1:\mathcal{K}_z}; \theta'\right) \tag{9}$$

where $t^l_{1:\mathcal{K}_z}$ represents the features of the last layer of the student model and $t^L_{1:\mathcal{K}_z}$ represents the features of the last layer of the teacher model, $T_{\theta'} : \mathfrak{X} \times \mathfrak{Y} \to \mathbb{R}$ is a neural network parameterized by θ' as in Section IV-A. In distillation training, the student model is trained with the weighted sum of $\mathcal{L}_D$

and the total loss employed during the training of the teacher model.

C. TATrack for UAV Tracking

TATrack is a target-aware tracking framework specifically designed for real-time UAV tracking, which leverages a target-aware ViT-based backbone, i.e., TA-ViT, coupled with a prediction head to improve the tracking performance. The framework aims to streamline the tracking process by simultaneously learning specialist features and conducting template–search coupling and to enhance target information in the feature representation. In this section, we depict the overall architecture of the TATrack and describe the prediction head and the total loss for training TATrack.

1) Overall Architecture: Based on TA-ViT, we build our TATrack, a compact end-to-end tracking framework for UAV tracking. Compared with other prevailing UAV trackers with separate processes of feature extraction and template–search coupling, it leads to a more compact and neat tracking pipeline with just a single backbone and a tracking head. The overall architecture is illustrated in Fig. 2. The input of TATrack 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 $\mathcal{K}_z = H_z W_z/P^2$ and $\mathcal{K}_x = H_x W_x/P^2$, respectively. Note that $\mathcal{K}_x$ is usually significantly larger than $\mathcal{K}_z$ to deal with large translation, which, therefore, gives rise to the problem as discussed in Section III. Given the feature representation $t^L_{1:\mathcal{K}}$ achieved by TA-ViT, the part corresponding to the search image (i.e., $t^L_{\mathcal{K}_z+1:\mathcal{K}}$) is supposed to have captured the correlation between the template and the search image. This part is subsequently fed into the prediction head for classification and regression tasks.

2) Prediction Head and Loss: Drawing inspiration from the corner detection head in [3] and [18], we utilize a prediction head $\mathcal{C}$ based on a fully convolutional network, comprising several convolutional-batch normalization-rectified linear unit (Conv-BN-ReLU) layers, for the direct estimation of the target's bounding box. The output tokens $t^L_{\mathcal{K}_z+1:\mathcal{K}}$, which correspond to the search image, are initially reinterpreted into a 2-D spatial feature map before being input into the prediction head. This results 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 initial estimation of the target position is determined by the maximum classification score, denoted by $(x_c, y_c) = \text{argmax}_{(x,y)}\mathbf{p}(x,y)$. The final target bounding box is then estimated based on this crude position by

$$\{(x_t, y_t); (w, h)\} = \{(x_c, y_c) + \mathbf{o}(x_c, y_c); \mathbf{s}(x_c, y_c)\}. \quad (10)$$

For the tracking task, we employ the weighted focal loss [54] for classification purposes and a mix of L_1 loss and GIoU loss [55] for bounding box regression. Finally, the overall loss function is

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{iou}}\mathcal{L}_{\text{iou}} + \lambda_{L_1}\mathcal{L}_{L_1} + \rho\mathcal{L}_{\text{MI}} \quad (11)$$

where the constants $\lambda_{\text{iou}} = 2$ and $\lambda_{L_1} = 5$ are set as in [3] and [18], ρ is set to 10^{-6}. Note that ρ is set to so small a value can be justified by the following reasons. Given that the target-awareness loss function $\mathcal{L}_{\text{MI}}$ as an auxiliary element to the primary tracking task, it is not intended to disproportionately influence the total loss $\mathcal{L}_{\text{overall}}$; hence, it is not set excessively high. In addition, $\mathcal{L}_{\text{MI}}$ often yields larger values compared with the other components, necessitating a relatively small weight to appropriately scale its impact. Finally, ρ is determined empirically. Our framework is trained end-to-end with the overall loss $\mathcal{L}_{\text{overall}}$ after the pretrained weights of the ViT for image classification are loaded. After this training, we employ the proposed framework of MI maximization-based knowledge-based distillation to obtain a student model that demonstrates a better trade-off between accuracy and efficiency. Specifically, during the distillation phase, we add the distillation loss $\mathcal{L}_D$ to the overall loss from the preceding training stage, resulting in the overall loss $\mathcal{L}^*_{\text{overall}}$ for distillation training as follows:

$$\mathcal{L}^*_{\text{overall}} = \mathcal{L}_{\text{overall}} + \sigma\mathcal{L}_D \quad (12)$$

where the balance coefficients in $\mathcal{L}_{\text{overall}}$ remain the same as in training the teacher model, and the weight σ of $\mathcal{L}_D$ is set to 5.

V. Experiments

In this section, our method is comprehensively evaluated on five well-known UAV tracking benchmarks, i.e., DTB70 [22], UAVDT [23], VisDrone2018 [24], UAV123 [25], and UAV123@10 FPS [25]. DTB70 [22] is a collection of 70 UAV sequences that, in addition to focusing on the problem of severe UAV motion, also includes various cluttered scenes and objects of varied sizes. UAVDT [23] is mainly used for vehicle tracking with various weather conditions, flying altitudes, and camera views. VisDrone2018 [24] is from a single object tracking challenge held in conjunction with the European Conference on Computer Vision (ECCV2018), it aims to evaluate drone tracking algorithms. UAV123 [25] is a large-scale aerial tracking benchmark involving 123 challenging sequences with more than 112k frames. UAV123@10 FPS [25] is in order to explore the effect of camera capture speed on tracking performance, which is created by sampling the UAV123 benchmark from the original 30–10 FPS. On a computer with an NVIDIA TitanX GPU, 16-GB RAM, and an i9-10850K processor (3.6 GHz), all assessment experiments are carried out. For a full comparison, 34 state-of-the-art trackers are used for comparison. Their results were obtained by running the official codes with the necessary hyperparameters. We separate them into two groups for a clearer comparison: 1) lightweight[1] trackers [2], [7], [8], [9], [10], [11], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68] and 2) deep trackers [3], [27], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80].

A. Implementation Details

In this section, we outline the key components and steps involved in the implementation of our tracking method. This

[1]The term "lightweight trackers" in this context refers to trackers that are either based on DCF or specifically developed for UAV tracking applications.

TABLE I

PRECISION (PREC.), SUCCESS RATE (SUCC.), AND SPEED (FPS) COMPARISON BETWEEN TATRACK AND LIGHTWEIGHT TRACKERS ON FIVE UAV TRACKING BENCHMARKS. RED, BLUE, AND GREEN INDICATE THE FIRST, SECOND, AND THIRD PLACES. NOTE THAT THE PERCENT SYMBOL (%) IS OMITTED FOR ALL PREC. AND SUCC. VALUES

	Method	Source	DTB70		UAVDT		VisDrone2018		UAV123		UAV123@10fps		Avg.		FLOPs	Param.	Avg. FPS	
			Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	(GMac)	(M)	GPU	CPU
DCF-based	KCF [56]	TPAMI 15	46.8	28.0	57.1	29.0	68.5	41.3	52.3	33.1	40.6	26.5	53.1	31.6	-	-	-	**622.5**
	SRDCFdecon [57]	CVPR 16	50.4	35.1	64.4	41.0	73.6	55.5	63.0	44.7	58.4	42.9	62.0	43.8	-	-	-	6.1
	fDSST [58]	TPAMI 17	53.4	35.7	66.6	38.3	69.8	51.0	58.3	40.5	51.6	37.9	60.0	40.7	-	-	-	**193.4**
	BACF [59]	ICCV 17	58.1	39.8	68.6	43.2	77.4	56.7	66.0	45.9	57.2	41.3	65.5	45.4	-	-	-	54.2
	ECO_HC [60]	CVPR 17	63.5	44.8	69.4	41.6	80.8	58.1	71.0	49.6	64.0	46.8	69.7	48.2	-	-	-	**83.5**
	STRCF [61]	CVPR 18	64.9	43.7	62.9	41.1	77.8	56.7	68.1	48.1	62.7	45.7	67.3	47.1	-	-	-	28.4
	MCCT_H [62]	CVPR 18	60.4	40.5	66.8	40.2	80.3	56.7	65.9	45.7	59.6	43.4	66.6	45.3	-	-	-	63.4
	ARCF [11]	ICCV 19	69.4	47.2	72.0	45.8	79.7	58.4	67.1	46.8	66.6	47.3	71.0	49.1	-	-	-	34.2
	AutoTrack [7]	CVPR 20	71.6	47.8	71.8	45.0	78.8	57.3	68.9	47.2	67.1	47.7	71.6	49.0	-	-	-	57.8
	RACF [63]	PR 22	72.5	50.5	77.3	49.4	83.4	60.0	70.2	47.7	69.4	48.6	74.6	51.2	-	-	-	35.6
CNN-based	SiamAPN [64]	ICRA 21	78.4	58.5	71.1	51.7	81.5	58.5	76.5	57.5	75.2	56.6	76.5	56.6	7.98	14.50	194.4	-
	SiamAPN++ [65]	IROS 21	78.9	59.4	76.9	55.6	73.5	53.2	76.8	58.2	76.4	58.1	76.5	56.9	8.23	14.72	167.5	-
	HiFT [8]	ICCV 21	80.2	59.4	65.2	47.5	71.9	52.6	78.7	59.0	74.9	57.0	74.2	55.1	7.32	10.42	160.3	-
	P-SiamFC++ [9]	ICME 22	80.3	60.4	80.7	56.6	80.1	58.5	74.5	48.9	73.1	54.9	77.7	55.9	2.15	10.44	240.5	46.1
	TCTrack [2] [10]	CVPR 22	81.2	62.2	72.5	53.0	79.9	59.4	80.0	60.5	78.0	59.9	78.3	59.0	8.77	9.75	139.6	-
	F-SiamFC++ [10]	IJCNN 22	81.4	60.5	79.4	55.5	80.7	59.6	78.9	59.2	72.1	54.5	78.5	57.9	1.83	10.23	**255.4**	51.6
ViT-based	LiteTrack [66]	ArXiv 23	82.5	**63.9**	81.6	**59.3**	79.7	61.4	84.2	65.9	**83.1**	64.0	82.2	62.9	6.78	26.18	119.7	-
	HiT [67]	ICCV 23	75.1	59.2	62.3	47.1	74.8	58.7	80.6	63.8	80.9	64.3	74.7	58.6	1.13	11.03	237.7	43.2
	MixFormer-V2 [68]	NIPS 23	73.7	57.0	57.8	42.1	69.1	52.9	81.3	63.7	81.0	63.6	72.6	55.9	4.40	16.04	184.6	37.2
	TATrack-ViT	**Ours**	82.2	**63.9**	80.5	58.6	**85.4**	65.0	**84.6**	**66.2**	**84.3**	**66.0**	83.4	63.9	2.39	8.08	203.3	47.7
	TATrack-PiT		82.3	63.6	80.8	58.0	82.9	63.4	81.4	64.1	79.8	63.4	81.4	62.5	1.08	6.94	**271.6**	55.3
	TATrack-XCiT		81.3	63.6	78.1	57.9	84.2	**65.5**	82.0	65.1	81.1	64.3	81.3	63.3	2.59	9.21	172.5	40.1
	TATrack-A-ViT		**84.8**	**65.9**	**81.9**	58.8	**85.6**	**65.4**	**84.7**	**66.9**	82.5	**65.5**	**83.9**	**64.5**	1.99	8.27	187.0	41.8
	TATrack-DeiT		**85.5**	**66.1**	**82.4**	**59.6**	**88.0**	**66.9**	**85.0**	**67.1**	**83.4**	**66.1**	**84.9**	**65.2**	2.39	8.27	242.3	49.1
	TATrack-DeiT-D		**85.0**	**65.9**	**83.4**	**60.6**	85.2	64.7	82.7	65.4	82.0	65.1	**83.7**	**64.3**	1.04	5.60	**320.7**	67.5

description aims to provide a clear understanding of the procedure and the elements used throughout the process. Our tracking framework is implemented in Python using PyTorch 1.9.0, with CUDA version 10.2. The model is trained and tested on a computer with an NVIDIA Titan X GPU.

1) Model: Our approach allows for the creation of various trackers using different ViT-based backbones. In this article, we use five efficient ViTs, including ViT-tiny [30], XCiT-tiny [81], DeiT-tiny [33], PiT-tiny [30], and A-ViT-tiny [82], for DeiT-tiny, we also take its first six layers as the ViT backbone of the student model for knowledge distillation training, denoted by DeiT-tiny-S. The above models are used as the backbone to construct six proposed trackers for evaluation. resulting in six trackers: TATrack-ViT, TATrack-DeiT, TATrack-PiT, TATrack-XCiT, TATrack-A-ViT, and TATrack-DeiT-D, respectively, where TATrack-DeiT-D is the student model associated with the teacher model TATrack-DeiT. The head of our model is a lightweight FCN, consisting of four stacked Conv-BN-ReLU layers for each of the three outputs. The sizes of the template and search image in all the proposed trackers are set to 128 $\times$ 128 and 256 $\times$ 256, respectively.

2) Training: The training pipeline is the same for all five trackers. The combination of the training sets from GOT-10k [83], LaSOT [84], COCO [85], and TrackingNet [86] are used for training. The batch size is 32. We use the AdamW optimizer [87] to train the model, and we set the weight decay to 10^{-4}, as well as the initial learning rate for the backbone to 4×10^{-5}. The number of training epochs is set to 300, with 60k image pairs for each epoch. After 240 epochs, the learning rate is reduced by a factor of 10. During distillation, we utilize the model trained in the previous stage as the teacher. The parameters of the teacher model are frozen to provide guidance to train the student with the proposed knowledge distillation, the training pipeline of which is the same as training the teacher model.

3) Inference: The Hanning window penalty is used during inference, in accordance with common practice [88], to impose positional prior on tracking. In more detail, the Hanning window of the same size is simply multiplied by the classification map, and the box with the greatest score after the multiplication is chosen as the tracking result.

B. Comparison With Lightweight Trackers

In this section, our TATrack is compared with 19 existing lightweight trackers on five UAV tracking benchmarks. The precision, success rate, and speed of the competing trackers on the five benchmarks are shown in Table I. We also provide a qualitative comparison between our method and state-of-the-art lightweight trackers.

1) Overall Performance Evaluation: The overall performance of our TATrack with the competing trackers on the five benchmarks is shown in Table I. It can be seen that our TATrack-* outperforms all other trackers on all benchmarks, in terms of average precision (Prec.) and success rate (Succ.). The highest average Prec. and Succ. among the DCF-based trackers are achieved by RACF [63], which are 74.6% and 51.2%, respectively. F-SiamFC++ [10] and TCTrack [2] attain the highest average Prec. and Succ. among the CNN-based trackers, with respective values of 78.5% and 59.0%. The lowest average Prec. and Succ. among the proposed ViT-based trackers are 81.3% and 62.5%, respectively, achieved by TATrack-XCiT and TATrack-PiT correspondingly. Note the apparent gaps between the highest performances of either DCF-based trackers or CNN-based trackers and the lowest ones of our methods. They are 6.7% and 11.3% over the DCF-based trackers, 2.8% and 3.5% over the CNN-based ones. Among other ViT-based trackers, LiteTrack [66] achieved the highest average Prec. and Succ., at 82.2% and 62.9%, respectively. But LiteTrack falls significantly behind our TATrack-DeiT by 2.7% and 2.3% in average Prec. and Succ., respectively, and is even 1.2% and 1.0% lower than the

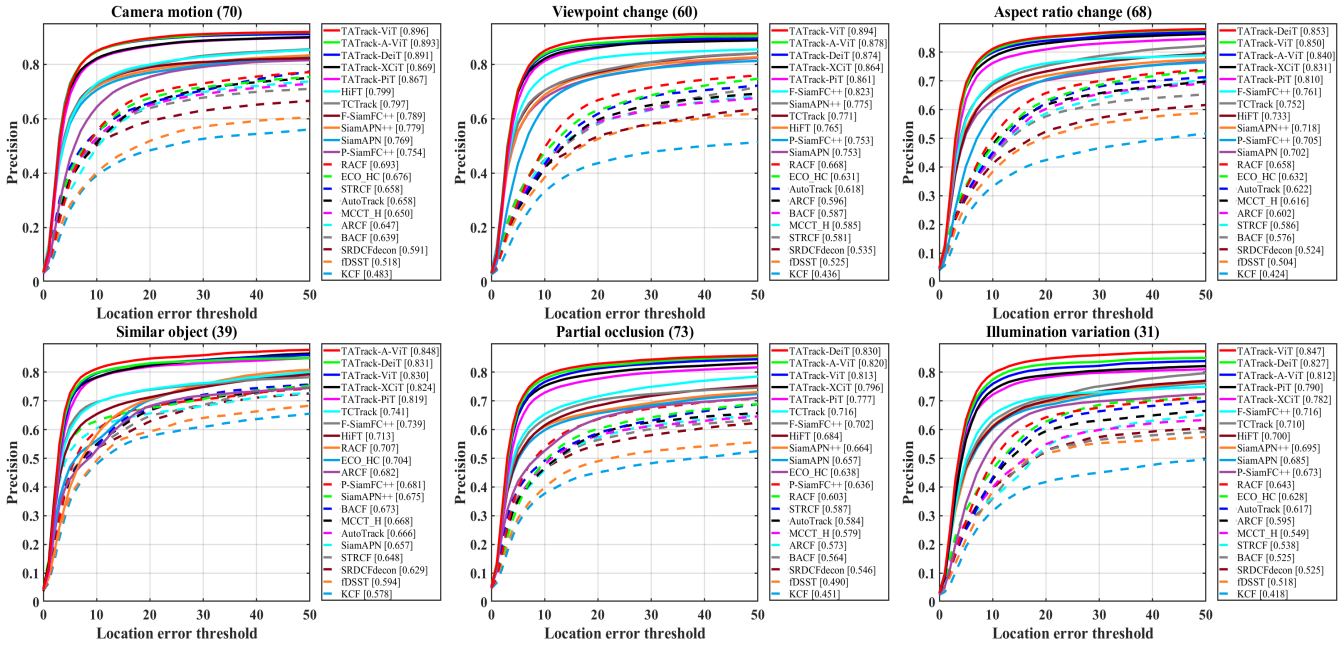


Fig. 4. Precision plots of attribute-based evaluation of all competing UAV trackers on UAV123 [25]. The precision at 20 pixels is used for ranking and marked in the precision plots. Our TATrack achieves top-five precision at all these attributes.

fourth-ranked TATrack-ViT of our trackers. More remarkable, the highest average Prec. and Succ. among the proposed trackers, which are achieved exclusively by TATrack-DeiT, are 84.9% and 65.2%, respectively, significantly surpassing those of DCF-based, CNN-based, and other ViT-based trackers. Specifically, the average Prec. gains are up to 10.3%, 6.4%, and 2.7%, respectively, while the gains of average Succ. are up to 14.0%, 6.2%, and 2.3%, respectively. In terms of GPU speed, the top-ranked and second-ranked trackers are the proposed TATrack-DeiT-D and TATrack-PiT, with respective 320.7 and 271.6 FPS. Although F-SiamFC++ [10] achieves the third place in GPU speed having 255.4 FPS, its average Prec. and succ. are significantly lower than ours. As for CPU speed, except for our TATrack-DeiT-D, all algorithms achieving above 60 FPS are among the DCF-based trackers, suggesting that the most efficient UAV trackers are still DCF-based. However, these fastest DCF-based trackers have much lower Prec. and Succ. than our methods. For example, KCF [56], the fastest one, gets only 31.6% in average Succ., about half of that of our methods. And, these DCF-based methods usually require a considerable cost to improve tracking precision. For example, RACF [63] is the best among DCF-based trackers in terms of Prec. and Succ., but it runs at only 35.6 FPS, obviously lower than the slowest CPU speed of our trackers. Despite the speeds of CNN-based and our ViT-based trackers are close, the Prec. and Succ. of the latter significantly outperform the former overall.

To provide a more compelling demonstration of the effectiveness of the proposed method, a comparative analysis of floating point operations per second (FLOPs) and Params. (number of parameters) based on DL-based approaches is also shown in Table I. As evident from the table, our method exhibits a relatively lower parameter count and reduced computational complexity when compared to state-of-the-art lightweight methods. The parameters of our trackers are fewer than those of all CNN-based ones. The FLOPs of all CNN-based trackers except P-SiamFC++ and F-SiamFC++ are above 7.0 GMac but those of our methods are below 3.0 GMac. For a more specific example, TATrack-PiT has 2.81 million fewer parameters than TCTrack and its FLOPs are only about one-eighth of TCTrack's. Nevertheless, our method achieves superior performance with an average precision and success rate that are 3.1% and 3.5% higher than TCTrack, respectively. These quantitative comparisons on efficiency further affirm the advantages of our approach compared with existing methods for real-time UAV tracking. In other ViT-based trackers, except for the FLOPs of HiT [67], the FLOPs and Params. of other methods are higher than ours. Our TATrack-PiT has similar FLOPs to HiT, but in comparison with TATrack-PiT, HiT demonstrates lower accuracy and speed. Remarkably, all the proposed ViT-based trackers can run at a real-time speed on a single CPU,[2] and the proposed TATrack-DeiT sets a new record of performance for real-time UAV tracking, justifying the effectiveness of the proposed methods.

2) Attribute-Based Evaluation: The proposed TATrack outperforms all the other DCF- and CNN-based UAV trackers in most attributes defined, respectively, in the five benchmarks. Examples of precision plots are shown in Fig. 4. As can be seen, in the situations of camera motion, viewpoint change, aspect ratio change, similar objects, partial occlusion, and illumination variation, TATrack considerably enhances its performance compared to other trackers. For example, our TATrack outperforms all the competing trackers in precision by more than 6.0% on camera motion, similar objects, partial

[2]Note that the real-time performance discussed in this article can be only generalized to platforms similar to or more advanced than ours.

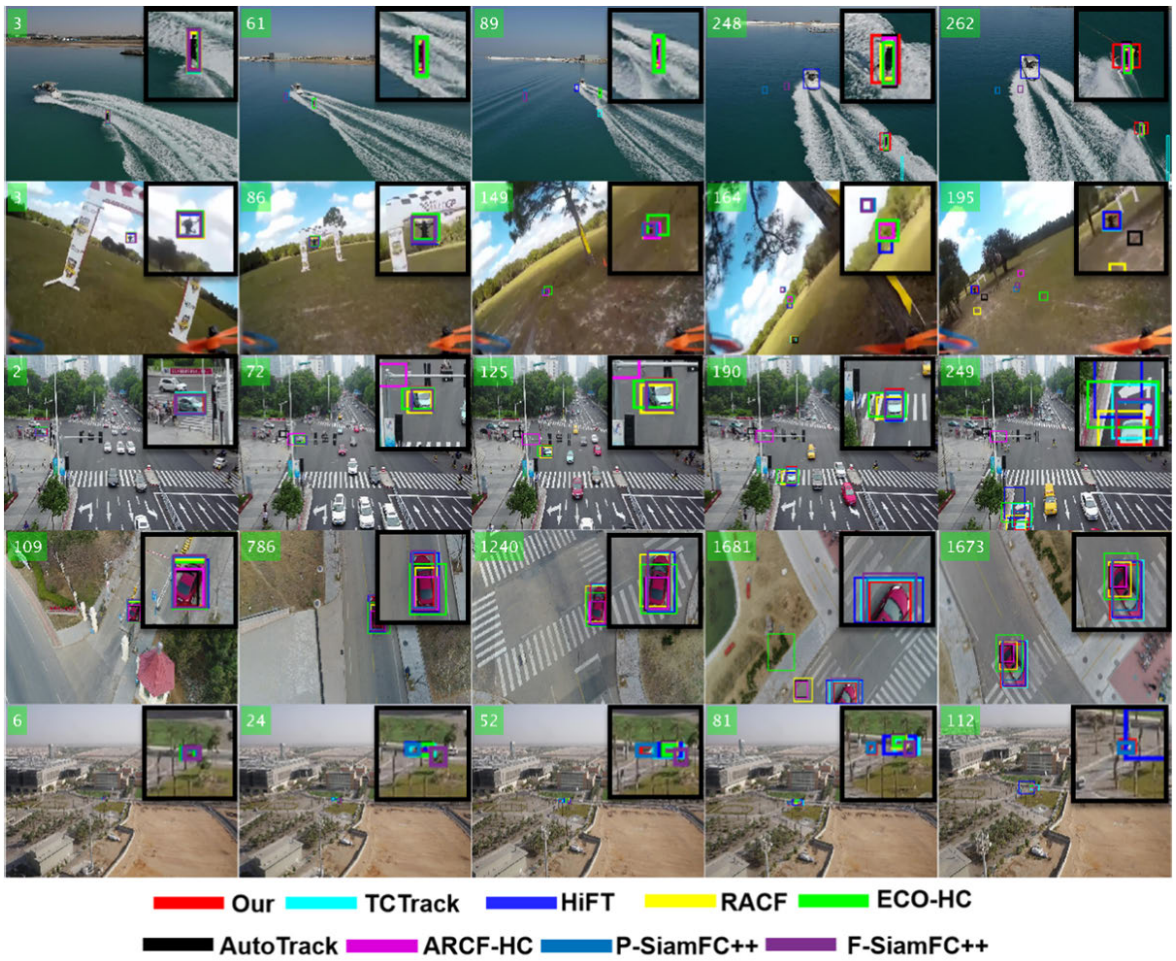


Fig. 5. Qualitative evaluation on five video sequences from, respectively, UAV123@10 FPS [25], DTB70 [22], VisDrone2018 [24], UAVDT [23], and UAV123 [25] (i.e., wakeboard3, ChasingDrones, uav0000294_00000_s, S1201, and car11).

TABLE II

COMPARISON BETWEEN TATRACK-DEIT AND DEEP-BASED TRACKERS ON FIVE UAV TRACKING DATASETS. RED, BLUE, AND GREEN INDICATE THE FIRST, SECOND, AND THIRD PLACES

Method	Source	DTB70		UAVDT		VisDrone2018		UAV123		UAV123@10fps		Avg. FPS
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	
TADT [27]	CVPR 19	69.2	47.4	67.0	42.0	71.7	51.8	71.2	52.7	66.6	50.6	**162.5**
SiamRPN++ [69]	CVPR 19	79.9	61.4	82.2	61.0	79.1	60.0	84	64.2	78.4	59.4	57.6
DiMP [70]	ICCV 19	79.2	61.3	78.3	57.4	83.5	63.0	85.6	65.4	85.1	64.7	51.3
PrDiMP [71]	CVPR 20	**84.0**	64.3	75.8	55.9	79.8	60.2	87.2	66.6	83.9	64.7	53.9
TrSiam [72]	CVPR 21	82.7	63.9	**88.9**	**65.0**	84.0	63.5	83.9	66.3	85.3	64.9	35.8
TransT [73]	CVPR 21	83.6	65.8	82.6	**64.2**	**85.9**	**65.2**	**87.6**	68.1	84.8	66.5	55.0
TrDiMP [72]	CVPR 21	82.4	63.9	**88.2**	**64.5**	84.1	63.1	87.2	67.5	**87.3**	66.5	35.8
AutoMatch [74]	ICCV 21	82.5	63.4	82.1	62.9	78.1	59.6	83.8	64.4	78.1	59.4	63.6
KeepTrack [75]	ICCV 21	83.6	64.3	83.8	60.5	84.0	63.5	**88.0**	**69.7**	**89.7**	**68.2**	20.3
CSWinTT [76]	CVPR 22	80.3	62.3	67.3	54.0	75.2	58.0	**87.6**	**70.5**	87.1	68.1	9.6
Tomp [77]	CVPR 22	**85.6**	**67.1**	**85.4**	64.1	84.1	64.4	85.0	67.8	**87.5**	67.9	23.8
SimTrack [78]	ECCV 22	83.2	64.6	76.5	57.2	80.0	60.9	**88.2**	**69.2**	**87.5**	**69.0**	72.6
OSTrack [3]	ECCV 22	82.7	65.0	85.0	63.4	84.2	**64.8**	84.7	67.4	83.1	66.1	68.3
SeqTrack [79]	CVPR 23	**85.6**	65.5	78.7	58.8	83.3	64.1	86.8	68.6	85.7	68.1	32.0
MAT [80]	CVRP 23	83.2	64.5	72.9	54.8	81.6	62.2	86.7	68.3	86.9	**68.5**	71.2
TATrack-DeiT	**Ours**	**85.5**	**66.1**	82.4	59.6	**88.0**	**66.9**	85.0	67.1	83.4	66.1	**242.3**
TATrack-DeiT-D		85.0	**65.9**	83.4	60.6	**85.2**	64.7	82.7	65.4	82.0	65.1	**320.7**

occlusion, and illumination variation. This validates the effectiveness of the proposed method in these challenging cases considered in the benchmarks.

3) Qualitative Evaluation: Some qualitative tracking results of TATrack and eight top trackers, i.e., TCTrack, HiFT, RACF, ECO-HC, AutoTrack, ARCF-HC, P-SiamFC++, and F-SiamFC++ are shown in Figs. 5 and 6. The former shows examples of our method performing better than other trackers, while the latter shows examples of our method failing to track. Figs. 5 and 6 each showcase five video sequences from five different benchmarks. For the former, the sequences include wakeboard3, ChasingDrones, uav0000294_00000_s, S1201, and car11. For the latter, the sequences are Animal2, uav_car2_s, uav8, S0501, and uav0000074_01656_s. In order

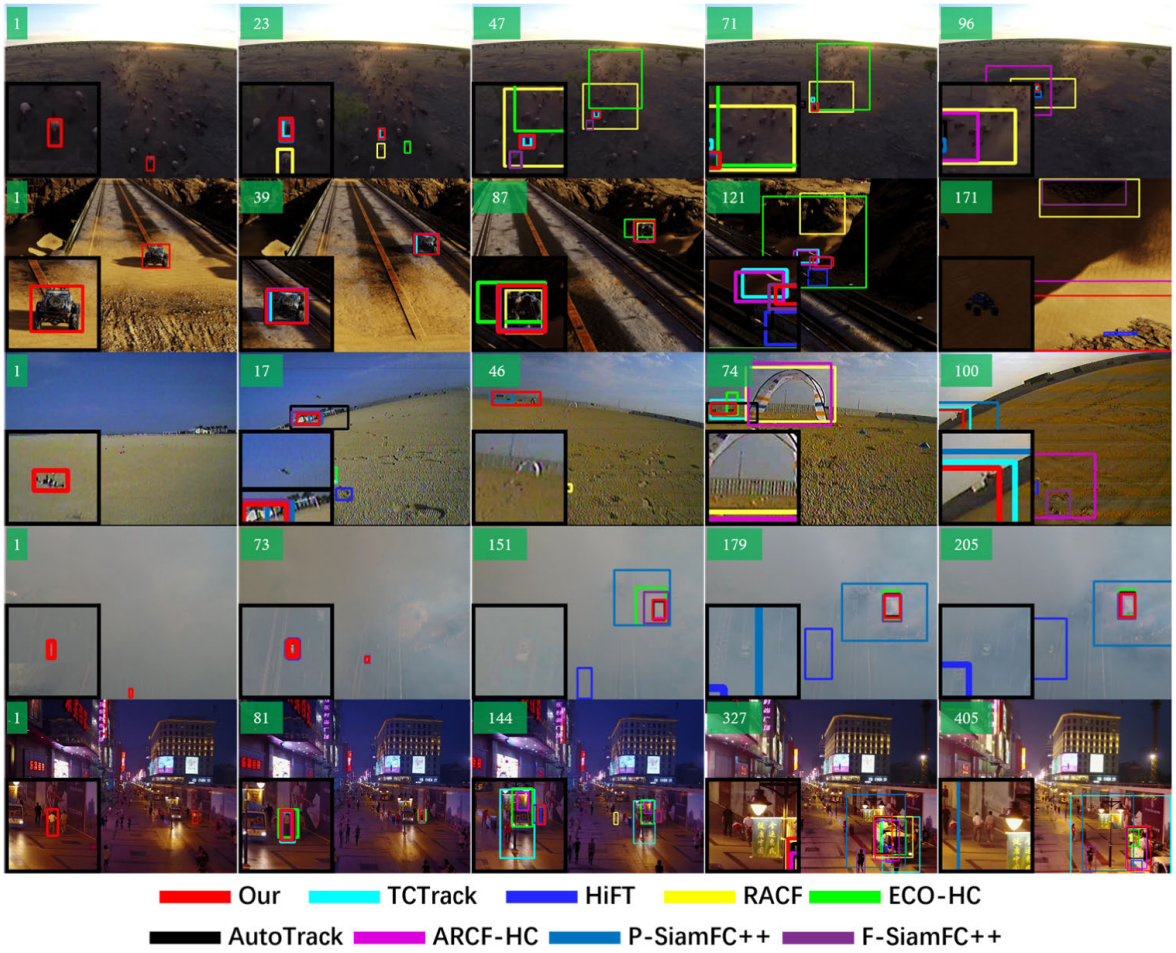


Fig. 6. Qualitative evaluation on five video sequences from, respectively, DTB70 [22], UAV123 [25], UAV123@10 FPS [25], UAVDT [23], and VisDrone2018 [24] (i.e., Animal2, uav_car2_s, uav8, S0501, and uav0000074_01656_s).

to provide clearer visualizations, targets appearing with low resolution have been suitably zoomed in and intercepted for display purposes. In Fig. 5, it can be observed that our tracker is the only one that successfully tracks the targets across all the challenging examples. These examples present various challenges, such as pose variations (i.e., in all sequences), background clusters (e.g., in uav0000294_00000_s), scale variations (e.g., in ChasingDrone, uav0000294_00000_s, and S1201). Our method significantly outperforms the others and provides more visually pleasing results in these cases. Specifically, only RACF, ECO-HC, AutoTrack, ARCF-HC, and our TATrack-DeiT succeed in tracking the target in wakeboard3 but TATrack-DeiT is more accurate; only TCTrack, HiFT, and TATrack-DeiT succeed in tracking the target in ChasingDrones; the target in car11 is successfully tracked by P-SiamFC++ and TATrack-DeiT only but TATrack-DeiT is more accurate; in the rest sequences, TATrack-DeiT is always the most accurate in tracking each target. These results demonstrate the superiority of the proposed method over these competing trackers. In Fig. 6, it is evident that all trackers eventually fail to maintain target tracking in these cases. Specifically, the first sequence features similar targets, making it difficult for the trackers to distinguish between them. The second sequence experiences severe illumination variations, challenging the trackers to maintain consistency in different lighting conditions. The third sequence involves fast motion and rapid viewpoint changes, testing the trackers' ability to adapt to quick and unpredictable movements. The fourth sequence deals with small objects and low resolution, where the limited detail makes accurate tracking more challenging. Finally, the last sequence encounters severe occlusion, requiring the trackers to predict the target's position even when it is partially or completely hidden from view. These cases illustrate the complexities of UAV tracking and highlight the limitations of both existing methods and our approach, suggesting the need for enhanced feature robustness, advanced motion, and appearance models, and better mechanisms for handling occlusion and illumination changes.

C. Comparison With Deep Trackers

In order to further highlight the strengths of the proposed method and showcase its superiority over other approaches in the field, the proposed TATrack-DeiT and TATrack-DeiT-D are also compared with 15 state-of-the-art deep trackers. The precision (Prec.), success rate (Succ.), and GPU speed of our TATrack-DeiT, TATrack-DeiT-D, and the competing deep trackers are shown in Table II. As can be seen, our proposed method, TATrack-DeiT, demonstrates superior performance compared to all other methods on the VisDrone2018

TABLE III

ABLATION STUDY OF WEIGHTING THE MI LOSS $\mathcal{L}_{\text{MI}}$ IN TRAINING TATRACK-DEIT, WITH ρ RANGING FROM 10^{-1} TO 10^{-7} WITH A SCALE FACTOR OF 0.1. THE EVALUATION IS CONDUCTED ON FIVE UAV TRACKING DATASETS. RED, BLUE, AND GREEN INDICATE THE FIRST, SECOND, AND THIRD PLACES

ρ	DTB70		UAVDT		VisDrone2018		UAV123		UAV123@10fps		Avg.	
	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
0	**85.2**	65.8	80.6	58.5	82.3	63.2	82.8	65.7	80.3	64.1	82.24	63.46
1×10^{-7}	**85.7**	**66.3**	80.3	58.2	84.9	64.5	82.4	65.3	82.4	**65.6**	**83.14**	**63.98**
1×10^{-6}	**85.5**	**66.1**	**82.4**	**59.6**	**88.0**	**66.9**	**85.0**	**67.1**	**83.4**	**66.1**	**84.86**	**65.16**
1×10^{-5}	84.8	65.4	79.6	58.0	**85.3**	**64.8**	**83.9**	**66.3**	81.8	65.0	83.08	**63.90**
1×10^{-4}	83.7	65.3	**83.2**	**59.4**	82.8	62.9	83.4	66.0	82.5	65.5	**83.12**	63.82
1×10^{-3}	84.5	65.2	82.1	58.9	**85.6**	**65.2**	82.1	65.1	81.1	64.6	83.08	63.80
1×10^{-2}	83.3	64.8	81.2	58.4	82.6	62.8	**84.4**	**66.6**	**83.1**	**66.0**	82.92	63.72
1×10^{-1}	84.5	**65.9**	**82.5**	**59.9**	82.4	62.7	82.3	65.2	**82.7**	**65.6**	82.88	63.86

benchmark. Specifically, it surpasses the second-place tracker, TransT, with improvements of 2.1% in precision and 1.7% in success rate. And, our method achieves second place on DTB70 in terms of both precision and success rate, with a small gap of 0.1% precision to the first-place Tomp and SeqTrack. These results show that our TATrack-DeiT is even comparable to state-of-the-art deep trackers in precision and success rates and underscore the effectiveness of our method for UAV tracking tasks. More remarkable, our TATrack-DeiT achieves the highest average GPU speed of 242.3 FPS, and TATrack-DeiT-D achieves an impressive 320.7 FPS, significantly outperforming all other methods. More specific, despite the precisions of Tomp, TrSiam SimTrack, and KeepTrack surpass our methods on DTB70, UAVDT, UAV123, and UAV123@10 FPS, respectively, these methods fall behind our methods in GPU speed apparently. Specifically, our TATrack-DeiT and TATrack-DeiT-D are, respectively, 9 and 12 times faster than Tomp, 5 and 8 times faster than TrSiam, 2 and 3 times faster than SimTrack, and 10 and 15 times faster than KeepTrack. This achievement underscores our method's ability to provide both high precision and speed, validating its suitability for UAV tracking that prioritizes efficiency as well as precision.

D. Ablation Study

In this section, an ablation study is performed to gain insight into the performance contributions of different components or characteristics of our proposed model.

1) Impact of Weighting the Loss for MI Maximization: To see how the weight ρ of the loss $\mathcal{L}_{\text{MI}}$ impacts the performance, we train TATrack-DeiT with different ρ that goes from 10^{-1} to 10^{-7} with a scale factor of 0.1 and evaluate them on VisDrone2018 [24]. The Prec. and Succ. are shown in Table III. Note that the baseline without target awareness corresponding to $\rho = 0$ is also shown in the table for comparison. As can be seen, the best Prec. (88.0%) and Succ. (66.9%) are achieved at $\rho = 10^{-6}$, while the second-ranked and third-ranked performances are achieved at $\rho = 10^{-3}$ and $\rho = 10^{-5}$, respectively. They apparently surpass the baseline performance Prec. (82.3%) and Succ. (63.2%) at $\rho = 0$, with the differences of 5.7% and 3.7%, respectively. Although Prec. increases for all $\rho > 0$ with respect to the baseline, there are some cases where Succ. slightly decreases, i.e., when $\rho = 10^{-1}, 10^{-2}$, and 10^{-4}. This suggests that the weight ρ does significantly impact the tracking performance and too large ρ may result in a negative effect. Only if appropriately weighted, will the MI loss lead to better tracking performance.

2) Impact of Architecture of the Jensen–Shannon MI Estimator: The neural network T_θ in the Jensen–Shannon MI estimator is composed of three linear layers. Its input size is fixed, dependent on the size of the input template and its feature representation, and its output size is 1. Therefore, the input and output size of T_θ is two hyperparameters. For simplicity, we set these two hyperparameters equal and denote them by l. To study how l impact the performance, we train TATrack-DeiT with different l that goes from 128 to 2048 with a scale factor of 2 and evaluate them on five UAV tracking datasets, i.e., DTB70 [22], UAVDT [23], VisDrone2018 [24], UAV123 [25], and UAV123@10 FPS [25]. The Prec. and Succ. are shown in Table IV. As can be seen, the best average Prec. and Succ. are achieved at $l = 512$, with the best average Prec. and Succ. being 84.86% and 65.16%, respectively. Basically, TATrack-DeiT achieves the best or the second-best Prec. or Succ. when l is between 256 and 1024, suggesting that l does significantly impact the tracking performance and only if appropriately set, will result in better tracking performance.

3) Effect of Learning Target-Aware ViTs: To evaluate the proposed idea of learning target-aware ViTs, the proposed trackers are also trained without the loss designed for learning target awareness and are evaluated on five UAV tracking benchmarks. The results are shown in Table V. The FLOPs and Params. (number of parameters) pertaining to inference are also shown in the table to help understand the computational complexity of the proposed trackers. As seen in the results, the Prec. and Succ. of all these trackers exhibit varying degrees of improvement when the proposed method for achieving target awareness is integrated. On average, the Prec. of TATrack-ViT, TATrack-XCiT, TATrack-DeiT, TATrack-PiT, and TATrack-A-ViT increases by 2.5%, 2.2%, 2.7%, 4.1%, and 1.6%, respectively, while their Succ. rises by 1.5%, 1.5%, 1.8%, 3.3%, and 1.3%, respectively. Although the enhancement of TATrack-PiT appears most noticeable, with 4.1% in Prec. and 3.3% in Succ., the improvements of the rest trackers are also significant when considering the following factors.

1) The mean growth in Prec. and Succ. for the CNN-based trackers, as derived from Table I, over the past two years is 2.4% and 1.4%, respectively.
2) The settings for learning occlusion-robustness and the training pipeline remain consistent across all trackers, without any customization.

TABLE IV

ABLATION STUDY OF THE ARCHITECTURE OF THE JENSEN–SHANNON MI ESTIMATOR. l IS THE SUPERPARAMETER THAT SPECIFIES THE NETWORK ARCHITECTURE. RED, BLUE, AND GREEN INDICATE THE FIRST, SECOND, AND THIRD PLACES

l	DTB70		UAVDT		VisDrone2018		UAV123		UAV123@10fps		Aug.	
	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
128	**85.3**	**65.6**	**81.6**	**58.8**	84.2	64.5	84.2	66.5	81.9	65.0	**83.44**	64.08
256	**84.8**	65.2	79.4	57.3	**86.1**	**65.8**	83.3	66.2	**82.9**	**66.0**	83.30	64.10
384	**85.3**	**65.8**	**82.6**	**59.0**	**85.2**	**64.7**	**85.2**	**67.1**	82.5	65.6	**84.16**	**64.44**
512	**85.5**	**66.1**	**82.4**	**59.6**	**88.0**	**66.9**	**85.0**	**67.1**	**83.4**	**66.1**	**84.86**	**65.16**
768	84.5	65.5	80.9	**59.0**	83.5	64.1	84.3	**66.8**	82.4	65.8	83.12	**64.24**
1024	83.3	64.4	80.0	58.2	81.3	62.0	**85.5**	**67.7**	**84.0**	**66.5**	82.82	63.76
2048	84.6	65.2	78.9	56.8	82.5	62.4	81.9	64.9	81.3	64.5	81.84	62.76

TABLE V

ABLATION STUDY ON EFFECT OF LEARNING TARGET-AWARE ViTs. THE PROPOSED TRACKERS ARE TRAINED WITH AND WITHOUT THE LOSS DESIGNED FOR LEARNING TARGET AWARENESS AND ARE EVALUATED ON FIVE UAV TRACKING BENCHMARKS

Model	target awareness	DTB70		UAVDT		VisDrone2018		UAV123		UAV123@10fps		Avg		FLOPs	Params.
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.		
TATrack-ViT	w/o	79.3	62.4	77.0	55.6	83.0	62.7	83.2	66.5	82.1	64.8	80.9	62.4	2.39G	8.08M
	w	**82.2**↑	**63.9**↑	**80.5**↑	**58.6**↑	**85.4**↑	**65.0**↑	**84.6**↑	66.2	**84.3**↑	**66.0**↑	**83.4**$_{2.5}$	**63.9**$_{1.5}$		
TATrack-XCiT	w/o	81.9	63.9	74.0	55.5	80.6	62.9	80.2	63.9	79.0	63.1	79.1	61.8	2.59G	9.21M
	w	81.3	63.6	**78.1**↑	**57.9**↑	**84.2**↑	**65.5**↑	**82.0**↑	**65.1**↑	**81.1**↑	**64.3**↑	**81.3**$_{2.2}$	**63.3**$_{1.5}$		
TATrack-DeiT	w/o	85.2	65.8	80.6	58.5	82.3	63.2	82.8	65.7	80.3	64.1	82.2	63.4	2.39G	8.27M
	w	**85.5**↑	**66.1**↑	**82.4**↑	**59.6**↑	**88.0**↑	**66.9**↑	**85.0**↑	**67.1**↑	**83.4**↑	**66.1**↑	**84.9**$_{2.7}$	**65.2**$_{1.8}$		
TATrack-PiT	w/o	79.2	61.8	76.2	54.4	76.2	58.2	78.4	61.5	76.6	60.8	77.3	59.3	1.08G	6.94M
	w	**82.3**↑	**63.6**↑	**80.8**↑	**58.0**↑	**82.9**↑	**63.4**↑	**81.4**↑	**64.1**↑	**79.8**↑	**63.4**↑	**81.4**$_{4.1}$	**62.6**$_{3.3}$		
TATrack-A-ViT	w/o	84.1	64.7	78.2	56.7	84.4	63.9	82.9	65.6	82.1	65.3	82.3	63.2	1.99G	8.27M
	w	**84.8**↑	**65.9**↑	**81.9**↑	**58.8**↑	**85.6**↑	**65.4**↑	**84.7**↑	**66.9**↑	**82.5**↑	**65.5**↑	**83.9**$_{1.6}$	**64.5**$_{1.3}$		

TABLE VI

EFFECT OF DIFFERENT KNOWLEDGE DISTILLATION LOSS FUNCTIONS WITH HIGHER ACCURACY BEING INDICATED IN **BOLD**

Method	KD loss	DTB70		UAVDT		VisDrone2018		UAV123		UAV123@10fps		Avg.		Avg. FPS
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	
TATrack-DeiT	-	85.5	66.1	82.4	59.6	88.0	66.9	85.0	67.1	83.4	66.1	84.9	65.2	242.3
TATrack-DeiT-D	MSE	83.6	64.4	81.6	58.9	79.9	61.0	82.5	65.3	79.7	63.5	81.5	62.6	318.9
	JSD	**85.0**	**65.9**	**83.4**	**60.6**	**85.2**	**64.7**	**82.7**	**65.4**	**82.0**	**65.1**	**83.7**	**64.3**	320.7

3) Our method can be easily integrated or adapted into other tracking frameworks, without increasing the inference time. As a final point, the consistent improvement observed in five UAV benchmarks, except for DTB70 [22], where TATrack-XCiT experiences a slight decrease, after employing the proposed method for learning target awareness, validates the effectiveness of our approach.

4) Effect of MI Maximization-Based Knowledge Distillation: To demonstrate the superiority of the proposed MI maximization-based knowledge distillation method, we use the MSE loss to replace the proposed MI-based loss $\mathcal{L}_{\text{MI}}$ to conduct the proposed feature-based knowledge distillation and evaluate the two approaches on five UAV tracking datasets. The experimental results are presented in Table VI. It can be observed from the table that our method consistently outperforms the approach using MSE loss across all five datasets, with particularly significant improvements on DTB70, UAVDT, and VisDrone2018. On average, employing $\mathcal{L}_{\text{MI}}$ leads to a 2.2% increase in Prec. and a 1.7% increase in Succ. compared with using MSE loss. In addition, compared with the teacher model, our method only incurs a slight decrease of 1.1% in Prec. and 0.9% in Succ., while achieving a notable speedup of 32%. These results highlight the advantage of our proposed MI maximization-based knowledge distillation method, which we attribute to the ability of MI-based loss to provide a more comprehensive measure of the relationship between features, allowing the student model to learn a richer representation from the teacher model. In addition, our MI-based loss is less sensitive to noise and outliers compared with MSE, making it particularly effective in noisy environments.

5) Application to Current Trackers: To demonstrate the generality of the proposed method, we applied it to three state-of-the-art trackers: GRM [89], HiT-Tiny [67], and OSTrack-256 [3]. We conducted a comparison of tracking performance with and without the integration of the target-aware component against the baseline, maintaining a consistent ρ across our tracking framework. The experimental results, as shown in Table VII, reveal significant improvements for all baseline methods when target awareness is incorporated. As can be seen, apart from a marginal decrease of 0.1% in GRM's Prec. on DTB70, all three baseline methods demonstrate performance enhancements on all five benchmarks after the integration of the proposed component. Specifically, GRM achieves a 3.0% improvement in Prec. and a 2.2% improvement in Succ. on UAV123 [25]. HiT-Tiny exhibits increases of 4.1% and 2.7% on VisDrone2018 [24], while OSTrack-256 demonstrates improvements of 3.4% and 2.4% on UAV123@10 FPS [25] in Prec. and Succ., respectively. These experimental results effectively highlight the versatility of the proposed method, which can be seamlessly integrated into existing tracking frameworks, enhancing tracking accuracy without incurring additional computational overhead.

6) Comparison Based on Consistent Training Datasets: As different DL-based trackers may be trained with different

TABLE VII

PROPOSED TARGET-AWARE METHODOLOGY WAS APPLIED TO SOTA TRACKERS, SPECIFICALLY GRM, HIT-TINY, AND OSTRACK-256, AND EVALUATED ON FIVE UAV TRACKING BENCHMARKS

Method	target awareness	DTB70		UAVDT		VisDrone2018		UAV123		UAV123@10fps		Avg	
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
GRM [89]	w/o	87.6	66.8	84.0	62.4	88.3	67.1	85.4	67.7	83.9	66.9	85.84	66.18
	w	87.5	**66.9**↑	**86.3**↑	**64.0**↑	**88.4**↑	**67.5**↑	**88.4**↑	**69.9**↑	**85.4**↑	**67.9**↑	**87.20**↑	**67.24**↑
HiT-Tiny [67]	w/o	64.7	51.3	54.2	41.4	64.6	50.3	74.1	58.7	74.8	59.2	66.48	52.18
	w	**68.6**↑	**54.1**↑	**59.4**↑	**43.6**↑	**68.7**↑	**53.0**↑	**78.1**↑	**61.9**↑	**78.2**↑	**61.9**↑	**70.60**↑	**54.90**↑
OSTrack-256 [3]	w/o	82.7	65.0	85.0	63.4	84.2	64.8	84.7	67.3	83.1	66.1	83.94	65.32
	w	**85.6**↑	**65.5**↑	**86.6**↑	**64.3**↑	**86.8**↑	**66.9**↑	**87.0**↑	**68.8**↑	**86.5**↑	**68.5**↑	**86.50**↑	**66.80**↑

TABLE VIII

ALL SIX DEEP LEARNING-BASED LIGHTWEIGHT SOTA TRACKERS WERE TESTED ON FIVE UAV TRACKING BENCHMARKS USING THE SAME TRAINING DATASET AS OUR METHOD

Method	DTB70		UAVDT		VisDrone2018		UAV123		UAV123@10fps		Avg	
	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
SiamAPN [64]	78.1	59.2	72.5	53.6	78.7	57.4	76.4	**59.0**	**74.0**	**56.9**	75.9	**57.2**
SiamAPN++ [65]	76.8	56.9	72.5	53.6	74.3	51.4	72.1	54.6	69.3	52.8	73.0	53.8
HiFT [8]	69.2	47.4	67.0	42.0	71.7	51.8	71.2	52.7	66.6	50.6	69.1	48.9
P-SiamFC++ [9]	80.1	60.2	**79.5**	**56.0**	**80.3**	**58.6**	74.7	49.2	73.3	55.0	**77.6**	55.8
TCTrack [2]	**82.8**	**63.2**	71.0	53.3	75.9	56.8	**76.5**	**60.3**	**74.1**	**59.0**	76.1	**58.5**
F-SiamFC++ [10]	**81.2**	**60.5**	**79.3**	**55.6**	**80.5**	**59.3**	**76.8**	55.2	72.8	54.6	**78.1**	57.0
TATrack-DeiT(Ours)	**85.5**	**66.1**	**82.4**	**59.6**	**88.0**	**66.9**	**85.0**	**67.1**	**83.4**	**66.1**	**84.9**	**65.2**

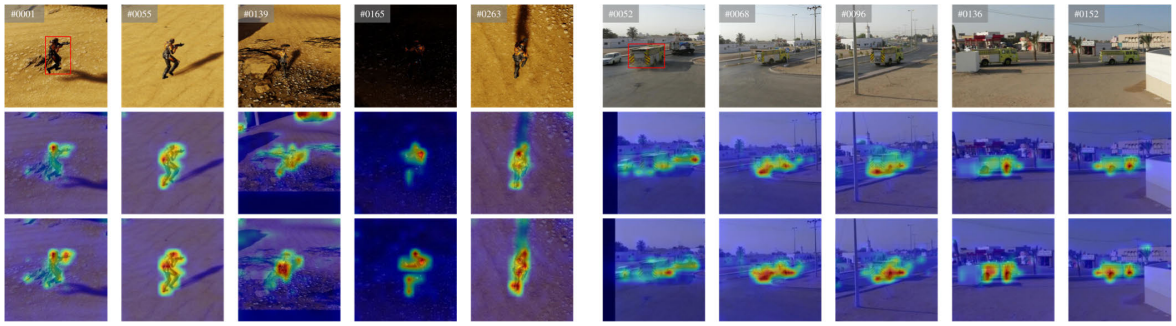


Fig. 7. For each group, the top shows the original images, while the following two rows display the attention map generated by the algorithm in some video frames. The middle row is from TATrack-DeiT-, and the bottom row is from TATrack-DeiT. Note that the suffix "-" is added to the model to indicate its lack of target awareness implementation.

datasets, in order to provide a comparison based on consistent training datasets, we trained all six state-of-the-art DL-based methods with the same datasets as in our training settings which are also adopted in many state-of-the-art trackers [3], [79], [80]. The six DL-based methods for comparison are SiamAPN [64], SiamAPN++ [65], HiFT [8], P-SiamFC++ [9], TCTrack [2], and F-SiamFC++ [10]. It is worth noting that all training codes are obtained from official sources, and modifications are limited to adapting the training set while keeping other aspects consistent with the original implementations. The experimental results are presented in Table VIII. As can be seen, even under the consistent training setting of identical datasets, our approach also consistently outperforms others. It excels in all five benchmarks, with several metrics demonstrating a substantial lead over the second-best approach. For instance, on VisDrone2018 [24], our TATrack-DeiT achieves a precision and success rate that is above 7.0% higher than the second-best tracker F-SiamFC++. On UAV123@10 FPS [25], our method surpasses the second-best, TCTrack, by an astonishing 9.3% in precision and 7.1% in success rate. On average, our method outperforms the second-best by 6.8% in precision and 6.7% in success rate. This significant performance advantage underscores the effectiveness of our approach. Notably, we observed that after training with the same dataset as ours, most of the six SOTA methods experienced a reduction in performance compared with their original performances. We speculate that this difference could be attributed to the fact that the hyperparameters in their methods were specifically tuned for the datasets they originally used and may not be optimal for the dataset we employed.

7) Qualitative Results: Figs. 7 and 8 visualize the attention produced by TATrack with and without target awareness implemented. For convenience, a suffix "-" is added to the models to indicate they do not have a target awareness implementation. For instance, TATrack-ViT- means TATrack-ViT without target awareness. Specifically, Fig. 8 aims to visualize the attention generated by TATrack with and without target awareness to illustrate the effectiveness of our approach in preserving and highlighting target information in learning efficient ViTs for UAV tracking, in which all examples are templates of the targets provided in the initial frame, which are less susceptible to occlusion to ensure clear identification of the target for tracking purposes. While Fig. 7 showcases attention maps for TATrack-DeiT and TATrack-DeiT on example sequences under more complex scenarios for comparison, where challenges

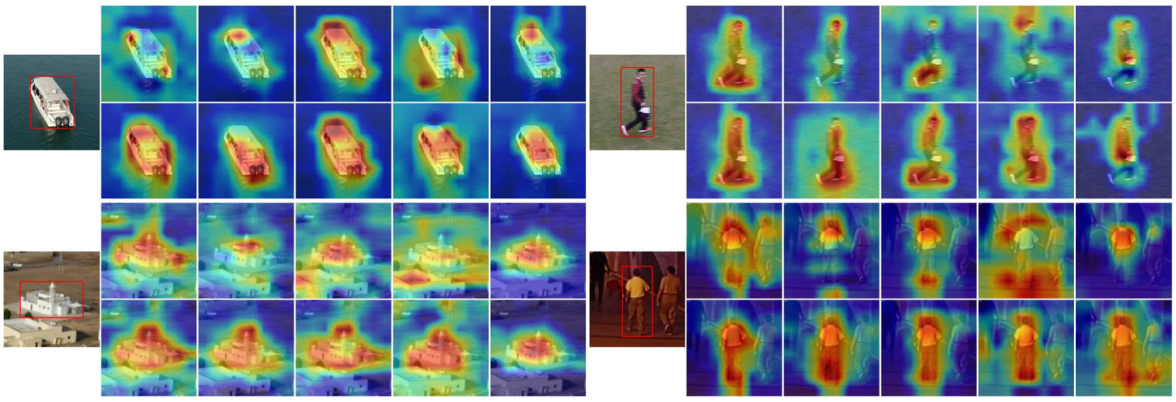


Fig. 8. For each group, we show (left) template, (top) attention, from left to right, generated, respectively, by TATrack-ViT-, TATrack-XCiT-, TATrack-DeiT-, TATrack-PiT-, and TATrack-A-ViT-, and (bottom) attention generated by TATrack-ViT, TATrack-XCiT, TATrack-DeiT, TATrack-PiT, and TATrack-A-ViT, respectively. Note that a suffix "-" is added to the model to indicate it does not have a target awareness implementation.

Fig. 9. Real-world tests. The tracking target has been marked with a red box in the real data recorded on the UAV platform.

such as background clutter, low light, aspect ratio changes, and occlusions are present. trackers equipped with target awareness yield more precise attention maps, particularly in complex scenarios. For instance, in scenes with background clutter and low light conditions, such as person1_s from the UAV123 dataset, TATrack-DeiT demonstrates sharper attention than TATrack-DeiT- even in low-contrast environments, clearly delineating the person's silhouette. Similarly, in scenes featuring aspect ratio variations and occlusions, like truck1 from the UAV123 dataset, TATrack-DeiT also exhibits more accurate attention. From Fig. 8, it can be seen that, without target awareness, generated attention either highlights only parts of a target (e.g., TATrack-DeiT-, TATrack-PiT- and TATrack-A-ViT- on ManRunning1 of DTB70 [22], TATrack-ViT-, TATrack-XCiT-, and TATrack-A-ViT- on boat2 of UAV123 [25]), or has a low contrast between the target and background (e.g., TATrack-XCiT- and TATrack-DeiT- on ManRunning1, TATrack-DeiT- and TATrack-PiT- on boat2). Whereas, when enhanced by target awareness, all the TATrack-* models generate visually more favorable attentions. For example, TATrack-ViT is able to generate attention that highlights most parts of the person and the boat, and the attention on the boat generated by TATrack-PiT has a higher contrast than by TATrack-PiT-. These qualitative results support the effectiveness of our method in maintaining and highlighting the target information in learning efficient ViTs for UAV tracking.

E. Real-World Tests

In this section, to verify the practicality of the method under real-world conditions, we installed an embedded onboard processor, the NVIDIA Jetson AGX, on a typical UAV platform. In real-world UAV testing, the utilization rates of GPU and CPU are 58.2% and 19.7%, respectively. The main challenge in the testing is shown in Fig. 9. The first line shows targets under strong sunlight, the second line of long-range small targets, and the third line targets with rapidly changing perspectives. Our tracker TATrack-DeiT succeeds in tracking all these targets. Moreover, our tracker remains at a speed of over 35.6 FPS during the tests without using TensorRT. Maintaining satisfactory tracking robustness in various challenging scenarios, real-world testing of TATrack-DeiT on embedded systems directly verifies excellent performance and efficiency in various UAV-specific challenges.

VI. Conclusion

In this study, we are the first to investigate the use of efficient ViTs within a unified template–search coupling

framework for real-time UAV tracking. We reveal and address an issue that target information is prone to diminish when performing template–search coupling with ViTs. Our method aims to learn target-aware ViTs by maximizing the MI between the template image and its feature representation produced by the ViT, which can be easily incorporated into other tracking frameworks without increasing inference time. Exhaustive experiments confirm the effectiveness of our method, demonstrating that our TATrack-DeiT establishes a new record in performance across five challenging UAV datasets. Building upon this, after applying our proposed MI maximization-based knowledge distillation, our TATrack-DeiT-D strikes a better trade-off between accuracy and efficiency, also showing state-of-the-art performance on the aforementioned five benchmarks, with a significant improvement in speed. We anticipate that our work will inspire further advancements in creating efficient ViT-based trackers for real-time UAV tracking.

Although we employed established ViTs for constructing our trackers in this study, we recognize that the efficiency of feature learning and template–search coupling is closely tied to these ViTs. Consequently, our future research endeavors will concentrate on exploring more lightweight and efficient ViTs. On the other hand, despite the fact that no additional inference time is required in achieving target-aware ViTs with the proposed method, MI estimation in our method necessitates a significant time cost for learning target-aware ViTs. In the future, we will investigate more efficient estimators for MI maximization in order to reduce training time. In addition, our method may fail due to the existence of similar targets, substantial variations in illumination, rapid and erratic movements, low resolution, and significant occlusion. This highlights the need for improved feature robustness, more advanced motion and appearance models, and better mechanisms for handling occlusion and illumination changes.

REFERENCES

[1] Z. Fu, Z. Fu, Q. Liu, W. Cai, and Y. Wang, "SparseTT: Visual tracking with sparse transformers," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 905–912.

[2] Z. Cao, Z. Huang, L. Pan, S. Zhang, Z. Liu, and C. Fu, "TCTrack: Temporal contexts for aerial tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14778–14788.

[3] B. Ye, H. Chang, B. Ma, S. Shan, and X. Chen, "Joint feature learning and relation modeling for tracking: A one-stream framework," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, Oct. 2022, pp. 341–357.

[4] S. Li, Y. Liu, Q. Zhao, and Z. Feng, "Learning residue-aware correlation filters and refining scale for real-time UAV tracking," *Pattern Recognit.*, vol. 127, Jul. 2022, Art. no. 108614.

[5] D. Yuan et al., "Active learning for deep visual tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 10, 2023, doi: 10.1109/TNNLS.2023.3266837.

[6] H. Zhang, W. Xing, Y. Yang, Y. Li, and D. Yuan, "SiamST: Siamese network with spatio-temporal awareness for object tracking," *Inf. Sci.*, vol. 634, pp. 122–139, Jul. 2023.

[7] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11920–11929.

[8] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "HiFT: Hierarchical feature transformer for aerial tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 15457–15466.

[9] X. Wang, D. Zeng, Q. Zhao, and S. Li, "Rank-based filter pruning for real-time UAV tracking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[10] W. Wu, P. Zhong, and S. Li, "Fisher pruning for real-time UAV tracking," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–7.

[11] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time UAV tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2019, pp. 2891–2900.

[12] C. Fu et al., "Siamese object tracking for unmanned aerial vehicle: A review and comprehensive analysis," *Artif. Intell. Rev.*, vol. 56, no. S1, pp. 1417–1477, Oct. 2023.

[13] X. Wang, X. Yang, H. Ye, and S. Li, "Learning disentangled representation with mutual information maximization for real-time UAV tracking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2023, pp. 1331–1336.

[14] D. Zeng, M. Zou, X. Wang, and S. Li, "Towards discriminative representations with contrastive instances for real-time UAV tracking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2023, pp. 1349–1354.

[15] S. Li, Q. Zhao, Z. Feng, and L. Lu, "Equivalence of correlation filter and convolution filter in visual tracking," in *Proc. 11th Int. Conf. Image Graph. (ICIG)*. Cham, Switzerland: Springer, 2021, pp. 623–634.

[16] M. Liu, Y. Wang, Q. Sun, and S. Li, "Global filter pruning with self-attention for real-time UAV tracking," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2022, p. 861.

[17] F. Xie, C. Wang, G. Wang, W. Yang, and W. Zeng, "Learning tracking representations via dual-branch fully transformer networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2688–2697.

[18] Y. Cui, C. Jiang, L. Wang, and G. Wu, "MixFormer: End-to-end tracking with iterative mixed attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13608–13618.

[19] F. Xie, C. Wang, G. Wang, Y. Cao, W. Yang, and W. Zeng, "Correlation-aware deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8741–8750.

[20] S. Li, Y. Yang, D. Zeng, and X. Wang, "Adaptive and background-aware vision transformer for real-time UAV tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 13989–14000.

[21] Y. Li, M. Liu, Y. Wu, X. Wang, X. Yang, and S. Li, "Learning adaptive and view-invariant vision transformer for real-time UAV tracking," in *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 1–18.

[22] S. Li and D. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4140–4146.

[23] D. Du et al., "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 370–386.

[24] L. Wen et al., "VisDrone-SOT2018: The vision meets drone single-object tracking challenge results," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 469–495.

[25] M. Mueller, N. G. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 445–461.

[26] J. Ye, C. Fu, G. Zheng, D. P. Paudel, and G. Chen, "Unsupervised domain adaptation for nighttime aerial tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8896–8905.

[27] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1369–1378.

[28] M. Guo et al., "Learning target-aware representation for visual tracking via informative interactions," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 927–934.

[29] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

[30] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[31] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 213–229.

[33] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 10347–10357.

[34] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 568–578.

[35] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 32–42.

[36] I. Tolstikhin et al., "MLP-mixer: An all-MLP architecture for vision," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24261–24272.

[37] W. Yu et al., "MetaFormer is actually what you need for vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10809–10819.

[38] R. Linsker, "Self-organization in a perceptual network," *Computer*, vol. 21, no. 3, pp. 105–117, Mar. 1988.

[39] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[40] J. P. Gou, B. S. Yu, S. J. Maybank, and D. C. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 31, pp. 1789–1819, Jul. 2021.

[41] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3967–3976.

[42] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1365–1374.

[43] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3048–3068, Jun. 2021.

[44] Z. Peng, Z. Li, J. Zhang, Y. Li, G. Qi, and J. Tang, "Few-shot image recognition with knowledge transfer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 441–449.

[45] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 742–751.

[46] Z. Yang et al., "Focal and global knowledge distillation for detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4643–4652.

[47] B. Li et al., "Learning light-weight translation models from deep transformer," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 13217–13225.

[48] L. Li, C. Chen, and X. Zhang, "Mask-guided self-distillation for visual tracking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[49] C. Sun, X. Wang, Z. Liu, Y. Wan, L. Zhang, and Y. Zhong, "SiamOHOT: A lightweight dual Siamese network for onboard hyperspectral object tracking via joint spatial–spectral knowledge distillation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5521112.

[50] S. Zhao, T. Xu, X.-J. Wu, and J. Kittler, "Distillation, ensemble and selection for building a better and faster Siamese based tracker," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 1, pp. 182–194, Mar. 2024.

[51] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10211–10221.

[52] B. Poole, S. Ozair, A. van den Oord, A. A. Alemi, and G. Tucker, "On variational bounds of mutual information," 2019, *arXiv:1905.06922*.

[53] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," 2018, *arXiv:1808.06670*.

[54] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 642–656, Mar. 2020.

[55] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[56] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[57] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1430–1438.

[58] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.

[59] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic Siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1781–1789.

[60] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

[61] F. Li, C. Tian, W. Zuo, L. Zhang, and M. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913.

[62] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4844–4853.

[63] S. Li, Y. Liu, Q. Zhao, and Z. Feng, "Learning residue-aware correlation filters and refining scale estimates with the GrabCut for real-time UAV tracking," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 1238–1248.

[64] C. Fu, Z. Cao, Y. Li, J. Ye, and C. Feng, "Onboard real-time aerial tracking with efficient Siamese anchor proposal network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5606913.

[65] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "SiamAPN++: Siamese attentional aggregation network for real-time UAV tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3086–3092.

[66] Q. Wei, B. Zeng, J. Liu, L. He, and G. Zeng, "LiteTrack: Layer pruning with asynchronous feature extraction for lightweight and efficient visual tracking," 2023, *arXiv:2309.09249*.

[67] B. Kang, X. Chen, D. Wang, H. Peng, and H. Lu, "Exploring lightweight hierarchical vision transformers for efficient visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 9612–9621.

[68] Y. Cui, T. Song, G. Wu, and L. Wang, "MixFormerV2: Efficient fully transformer tracking," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2024, pp. 58736–58751.

[69] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4277–4286.

[70] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6182–6191.

[71] M. Danelljan, L. Van Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7181–7190.

[72] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1571–1580.

[73] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8122–8131.

[74] Z. Zhang, Y. Liu, X. Wang, B. Li, and W. Hu, "Learn to match: Automatic matching network design for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13319–13328.

[75] C. Mayer, M. Danelljan, D. P. Paudel, and L. Van Gool, "Learning target candidate association to keep track of what not to track," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13424–13434.

[76] Z. Song, J. Yu, Y. P. Chen, and W. Yang, "Transformer tracking with cyclic shifting window attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8781–8790.

[77] C. Mayer et al., "Transforming model prediction for tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8721–8730.

[78] B. Chen et al., "Backbone is all your need: A simplified architecture for visual object tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 375–392.

[79] X. Chen, H. Peng, D. Wang, H. Lu, and H. Hu, "SeqTrack: Sequence to sequence learning for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 14572–14581.

[80] H. Zhao, D. Wang, and H. Lu, "Representation learning for visual object tracking by masked appearance transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 18696–18705.

[81] A. Ali et al., "XCiT: Cross-covariance image transformers," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 34, Dec. 2021, pp. 20014–20027.

[82] H. Yin, A. Vahdat, J. M. Alvarez, A. Mallya, J. Kautz, and P. Molchanov, "A-ViT: Adaptive tokens for efficient vision transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 10809–10818.

[83] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021.

[84] H. Fan et al., "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5369–5378.

[85] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[86] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 310–327.

[87] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arxiv:1711.05101*.

[88] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," 2020, *arXiv:2006.10721*.

[89] S. Gao, C. Zhou, and J. Zhang, "Generalized relation modeling for transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 18686–18695.

Shuiwang Li (Member, IEEE) received the Ph.D. degree in computer science and technology from Sichuan University, Chengdu, China, in 2021.

In 2015, he joined the Institute of Computational and Theoretical Studies, Hong Kong Baptist University, Hong Kong, as a Research Assistant. He is currently an Assistant Professor with the College of Computer Science and Engineering, Guilin University of Technology, Guilin, China. His research interests include pattern recognition, computer vision, and machine learning.

Xiangyang Yang received the B.Sc. degree in network engineering from Tiangong University, Tianjin, China, in 2019. He is currently pursuing the master's degree with the College of Computer Science and Engineering, Guilin University of Technology, Guilin, China.

His research interests are in computer vision, especially in object tracking.

Xucheng Wang is currently purusing the bachelor's degree with the College of Computer Science and Engineering, Guilin University of Technology, Guilin, China.

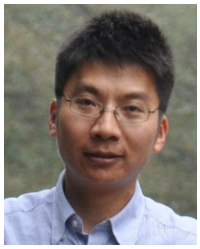
His research interests are in pattern recognition and computer vision, especially in object tracking.

Dan Zeng received the B.E. and Ph.D. degrees in computer science and technology from Sichuan University, Chengdu, China, in 2013 and 2018, respectively.

From 2018 to 2020, she was a Post-Doctoral Research Fellow with the Data Management and Biometrics Group, University of Twente, Enschede, The Netherlands. She is currently a Research Associate Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. Her main research topics include biometrics, computer vision, and deep learning.

Hengzhou Ye received the B.S. degree from Guilin University of Technology, Guilin, China, in 2002, and the Ph.D. degree from Guangxi University, Nanning, China, in 2019.

He is currently a Full Professor at Guilin University of Technology. His research interests include mobile edge computing, multiobjective optimization, object detection, and object tracking.



Qijun Zhao received the B.Sc. and M.Sc. degrees in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science from The Hong Kong Polytechnic University, Hong Kong, in 2010.

He was a Post-Doctoral Research Fellow with the Pattern Recognition and Image Processing Laboratory, Michigan State University, East Lansing, MI, USA, from 2010 to 2012. He is currently a Professor with the College of Computer Science, Sichuan University, Chengdu, China. He is also a Visiting Professor with the School of Information Science and Technology, Tibet University, Lhasa, China. His research interests in the fields of pattern recognition, image processing, and computer vision.