# Learning Adaptive and View-Invariant Vision Transformer for Real-Time UAV Tracking

**Yongxin Li** [*1] **Mengyuan Liu** [*1] **You Wu** [*1] **Xucheng Wang** [1] **Xiangyang Yang** [1] **Shuiwang Li** [1]

## Abstract

Harnessing transformer-based models, visual tracking has made substantial strides. However, the sluggish performance of current trackers limits their practicality on devices with constrained computational capabilities, especially for real-time unmanned aerial vehicle (UAV) tracking. Addressing this challenge, we introduce AVTrack, an adaptive computation framework tailored to selectively activate transformer blocks for real-time UAV tracking in this work. Our novel Activation Module (AM) dynamically optimizes ViT architecture, selectively engaging relevant components and enhancing inference efficiency without compromising much tracking performance. Moreover, we bolster the effectiveness of ViTs, particularly in addressing challenges arising from extreme changes in viewing angles commonly encountered in UAV tracking, by learning view-invariant representations through mutual information maximization. Extensive experiments on five tracking benchmarks affirm the effectiveness and versatility of our approach, positioning it as a state-of-the-art solution in visual tracking. Code is released at: https://github.com/wuyou3474/AVTrack.

## 1. Introduction

As unmanned aerial vehicles (UAVs) continue to evolve and diversify in their applications, the field of UAV tracking has become increasingly critical. UAV tracking involves assessing and predicting the positions of arbitrary targets within continuous aerial imagery, and presents unique challenges, including handling extreme view angle, mitigating motion blur, and overcoming severe occlusion. In addition,

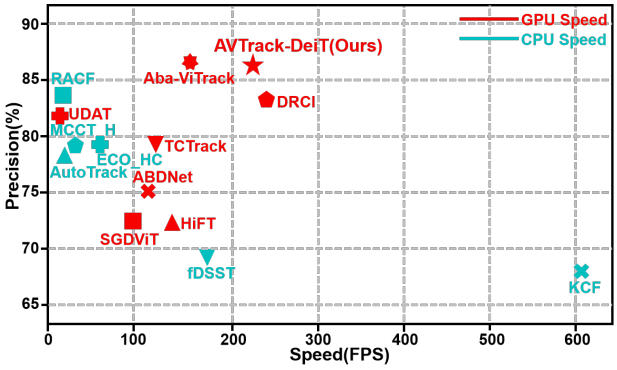


*Figure 1.* Compared with state-of-the-art UAV tracking algorithms on VisDrone2018. Our AVTrack-DeiT sets a new record with 0.860 precision and still runs efficiently at approximately 220 FPS.

the efficiency of UAV tracking is paramount, given the limited battery capacity and computing resources inherent to UAVs (Li et al., 2023; Wang et al., 2023; Ma et al., 2023; Liu et al., 2022a; Li et al., 2020b). Researchers and practitioners are actively exploring innovative approaches to refine tracking algorithms, accuracy, and efficiency to meet the unique demands of UAV applications.

Although discriminative correlation filters (DCF)-based trackers dominate the filed of UAV tracking for their high efficiency, they hardly match deep learning (DL)-based methods in tracking precision. Especially notable is the growing trend in the DL-based methods towards the adoption of single-stream architectures, seamlessly integrating feature extraction and fusion through pre-trained Vision Transformer (ViT) backbone networks. The success of recent approaches like OSTRack (Ye et al., 2022a), SimTrack (Chen et al., 2022), Mixformer (Cui et al., 2022), and DropMAE (Wu et al., 2023) exemplifies the effectiveness of this paradigm shift. Inspired by this, Aba-VTrack (Li et al., 2023) delivers an lightweight DL-based tracker based on this framework with an adaptive and background-aware token computation method to reduce inference time, and shows remarkable precision and speed for real-time UAV tracking. Nonetheless, the utilization of a variable number of tokens in this approach leads to a significant time cost, primarily arising from unstructured access operations.

[*]Equal contribution [1]College of Computer Science and Engineering, Guilin University of Technology, Guilin, China. Correspondence to: Shuiwang Li <lishuiwang0721@163.com>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

In this study, we also adopt a single-stream architecture driven by pre-trained transformer backbone networks. However, our emphasis lies in improving the efficiency of ViTs through more structured methods. To this end, we introduce an activation module into each transformer block. The output of this module is an activation probability, which determines if a transformer block should be activated for a certain given input. As the ViT is adaptively trimmed by blocks, time-consuming unstructured access operations are avoided in our approach. The rationale behind is grounded in the recognition that semantic features or relations do not uniformly impact the tracking task across all abstraction levels. Instead, this impact varies based on the characteristics of the target and the scene it occupies. For example, when a target moves against a monochromatic background, effective tracking can be achieved based on the color contrast between the target and the background, owing to the simplicity of the scene. In such cases, this straightforward feature is often sufficient. However, tracking tasks become more intricate when the target navigates through a cluttered environment. In cluttered scenarios, capturing and analyzing sufficient semantic features and relations become crucial for achieving effective tracking. This highlights the dynamic nature of tracking demands, intricately tied to the specific characteristics of the scene and the target being tracked. In our implementation, the activation module is a linear layer followed by a nonlinear activation function, which takes only a slice of all tokens (representing both the target template and the search image) as input in consideration of efficiency. By tailoring the architecture of ViTs to the specific demands of a tracking task, our approach holds the potential to accelerate the inference process for visual tracking.

Additionally, to enhance the effectiveness of ViTs for UAV tracking, we introduce a novel approach to learn view-invariant feature representations. This is achieved by maximizing the mutual information between the backbone features extracted from two different views of the target. Notably, to the best of our knowledge, this perspective on feature learning has not been extensively explored in the context of UAV tracking. Mutual information is a measure that quantifies the dependence or relationship between two variables (Steuer et al., 2002). Mutual information maximization refers to the process of enhancing the mutual information between different components or variables within a system, which has found wide applications in various computer vision tasks (Liu et al., 2022b; Yang et al., 2022; R.D. & et al., 2019). By maximizing mutual information of two different views of the target, we aim to ensure that the learned representations preserve essential information about the target regardless of changes in viewpoint. We refer to so obtained representations as view-invariant representations. We believe models trained with view-invariant representations tend to generalize better across diverse viewing conditions, making them more robust in real-world scenarios where changes in viewpoint are common. The advantages of such view-invariant representations become especially pronounced in the context of UAV tracking, where the challenges of extreme changes in viewing angles are prevalent. Furthermore, since the template image and the target patch in the search image represent two different views of the same target, our method can be seamlessly integrated into existing tracking frameworks with the incorporation of just an additional loss. We call this proposed adaptive computation framework AVTrack. Extensive experiments substantiate the effectiveness, efficiency and generality of our method and demonstrate that our AVTrack achieves state-of-the-art real-time performance. As shown in Fig. 1, our method sets a new record with a precision of 0.860 and runs efficiently at around 220 frames per second (FPS) on the VisDrone2018. Our primary contributions can be summarized as follows:

- In view of that the tracking process is more efficient if it operates as a dynamic and context-sensitive mechanism, we propose the Activation Module to adaptively activate transformer blocks for real-time UAV tracking based on ViTs.
- We propose to learn view-invariant feature representations by maximizing the mutual information between the backbone features of two different views of the target. This approach results in more effective and informative feature representations, particularly addressing challenges arising from changes in viewing angles.
- We introduce AVTrack, a family of efficient trackers based on these components, which integrates seamlessly with other ViT-based trackers. AVTrack demonstrates promising performance while maintaining extremely fast tracking speeds. Empirical evaluations show that AVTrack achieves state-of-the-art real-time performance.

## 2. Related Works

### 2.1. Visual Tracking

There are two main types of modern visual trackers: DCF-based and DL-based trackers (Zhang et al., 2022b; Li et al., 2020a; 2021a;b; Zhong et al., 2022; Liu et al., 2022a). DCF-based trackers are known for their high efficiency on CPUs. However, they face challenges in maintaining robustness in challenging conditions due to limited feature representation capabilities. Lightweight DL-based trackers, like those in (Cao et al., 2021; 2022), show advancements in tracking precision and robustness for UAV tracking, but their efficiency lags behind DCF-based trackers considerably. Although model compression techniques, as seen in (Wang et al.,

2022; Wu et al., 2022; Zhong et al., 2023), were utilized to enhance efficiency, these trackers still face challenges associated with unsatisfactory tracking precision. Vision Transformers (ViTs) are becoming prominent for streamlining and unifying frameworks in generic visual tracking, as evident in studies like (Xie et al., 2021; Cui et al., 2022; Ye et al., 2022a; Xie et al., 2022). Notable ones include a Siamese-style dual-branch network (Xie et al., 2021), a Transformer-based Mixed Attention Module (MAM) (Cui et al., 2022), and a single-stream tracking framework featuring an elimination module for in-network candidates (Ye et al., 2022a). Although these frameworks are efficient owing to their compact nature, very few of them are based on lightweight ViTs, rendering them impractical for real-time UAV tracking. In an effort to overcome this limitation, Aba-ViTrack (Li et al., 2023) utilized lightweight ViTs and implemented an adaptive, background-aware token computation method to improve efficiency for real-time UAV tracking. However, the variable token number in this approach necessitates unstructured access operations, leading to notable time costs. In this work, we focus on enhancing the efficiency of ViTs through more structured methods for UAV tracking.

### 2.2. Efficient Vision Transformer

Efforts to improve the efficiency of ViTs have attracted considerable attention in recent research, reflecting the necessity to balance their representation capabilities with computational efficiency. Methods include the development of lightweight ViTs utilizing low-rank methods, model compression, and hybrid designs (Wang et al., 2020; Zhang et al., 2022a; Mao et al., 2021; Chen et al., 2021b; Li et al., 2022b). However, ViTs designed with low-rank and quantization methods often sacrifice considerable accuracy to achieve efficiency gains. Pruning-based ViTs usually necessitate careful decisions regarding pruning ratios and entail a time-consuming fine-tuning process. Additionally, hybrid ViTs incorporating CNN-based stems impose restrictions on input size flexibility.

With growing popularity, recent developments in efficient ViTs with conditional computation focus on adaptive inference for model acceleration. This approach dynamically adjusts the computational load based on input complexity, allowing ViTs to allocate resources judiciously during inference. For example, DynamicViT (Rao et al., 2021) introduces control gates to selectively process tokens, while A-ViT (Yin et al., 2022) employs an Adaptive Computation Time strategy to avoid auxiliary halting networks, achieving gains in efficiency, accuracy, and token prioritization. The latter is also exploited in Aba-VTrack (Li et al., 2023) to built efficient trackers for real-time UAV tracking. However, the use of a variable number of tokens incurr significant time costs due to additional unstructured access operations. In this study, we explore the adaptive activation of specific Transformer blocks for feature representation, a more structured and effective form of conditional computation, to improve the efficiency of ViTs.

### 2.3. View-Invariant Feature Representation

View-invariant feature representation has garnered significant attention in the field of computer vision and image processing. This technique aims to extract features from images or visual data that remain consistent across various viewpoints or orientations, providing robustness to changes in the camera angle or scene configuration (Kumie et al., 2024; Bracci et al., 2018; Li et al., 2017; Rao et al., 2002). Early efforts in achieving view-invariant representations often relied on handcrafted features and geometric transformations (Xia et al., 2012; Ji & Liu, 2010; Rao et al., 2002). These methods, while effective in certain scenarios, struggled to handle the complexity and variability inherent in real-world visual data. The advent of deep learning revolutionized the field. Many studies have explored the application of Convolutional Neural Networks (CNNs) and other deep architectures for extracting view-invariant representations (Kumie et al., 2024; Gao et al., 2022; Shiraga et al., 2016). These approaches leverage the capacity of deep models to capture complex patterns and variations in visual data. By ensuring that learned features are resilient to changes in viewpoint, these methods contribute to the robustness and generalization of vision-based systems. View-invariant feature representation has proven valuable across a spectrum of computer vision applications, such as action recognition (Kumie et al., 2024), pose estimation (Bracci et al., 2018), and object detection (Feng et al., 2022). Despite its efficacy in these domains, the exploration of view-invariant feature representations in the context of visual tracking remains limited. Notably, there is a dearth of research on integrating view-invariant representations into visual tracking frameworks, to the best of our knowledge. In this study, we embark on the exploration of learning view-invariant feature representations with mutual information maximization based on ViTs, specifically tailored for UAV tracking. This marks the first instance where ViTs are employed for the purpose of acquiring view-invariant feature representations in the context of UAV tracking.

## 3. Method

In this section, we first provide a brief overview of our end-to-end tracking framework, named AVTrack, as shown in Fig. 2. Then, we introduce the Activation Module (AM) for dynamically activating transformer blocks based on inputs and the method for learning View-Invariant Representations (VIR) via mutual information maxmization. Finally, we detail the prediction head and training loss.

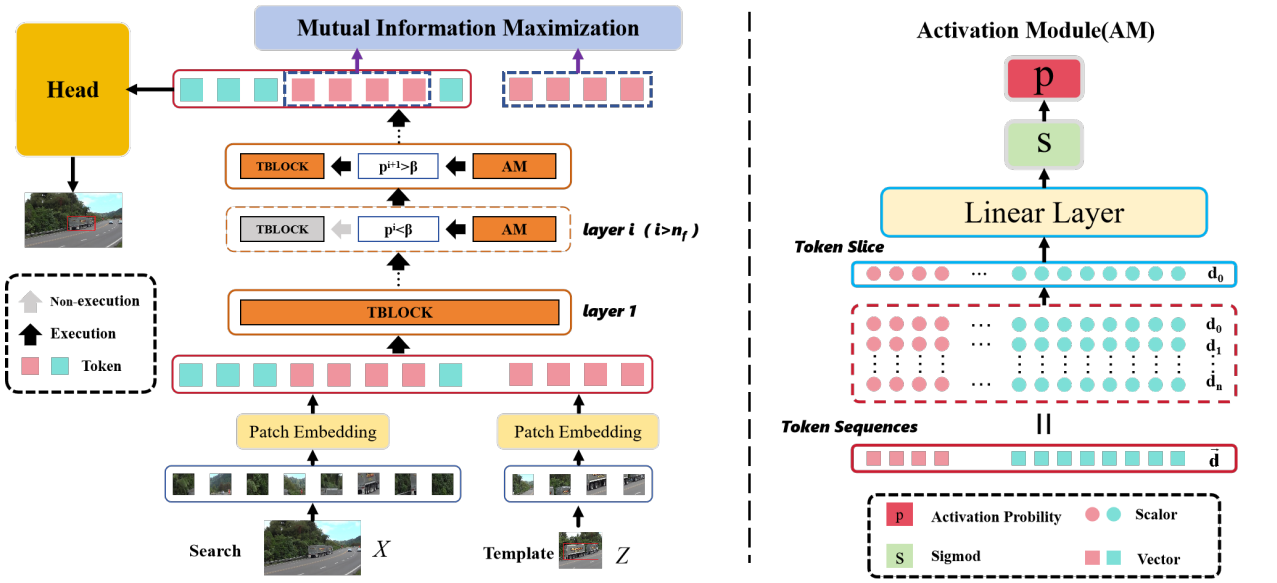


*Figure 2.* (left) Overview of the proposed AVTrack's framework, which consists of a single-stream backbone and a prediction head. Activattion Modules (AMs) are added into Transformer blocks (TBlocks) to make the ViT adaptive and Mutual Information Maximization is employed to learn View-Invariance Representations (VIR). (right) The detailed structure of the Activation Module (AM).

### 3.1. Overview

The proposed AVTrack introduces an novel single-stream tracking framework, featuring an adaptive ViT-based backbone and a prediction head. Each transformer block except the first $n_f$ ones within the ViT-based backbone incorporates an Activation Module (AM), which is trained to adaptively decide whether to activate the associated transformer block or not to enhance the efficient of the ViT. The framework takes a pair of images as input, comprising a template denoted as $Z \in \mathbb{R}^{3\times H_z\times W_z}$ and a search image denoted as $X \in \mathbb{R}^{3\times H_x\times W_x}$. These images are split into patches of size $P \times P$, and the number of patches for $Z$ and $X$ are $P_z = H_z \times W_z/P^2$ and $P_x = H_x \times W_x/P^2$, respectively. The features extracted from the backbone are input into the prediction head to generate tracking outcomes. To obtain view-invariant representation with ViTs, we maximize the mutual information between the feature representations of two different views of the target, i.e., the template image and the target patch in the search image. During the training phase, since the ground truth localization of the target in the search image is known, we can obtain the feature representation of the subsequent view from the representation of the search image using interpolation techniques. The details of the these components will be elaborated in the subsequent subsections.

### 3.2. Activation Module (AM)

The Activation Modules dynamically activates transformer blocks based on inputs, allowing adaptive adjustments to the ViT's architecture, hopefully activating only the necessary blocks to successfully perform the tracking task. AM takes a slice of all tokens representing both the target template and the search image as input for efficiency considerations. Its output indicates the probability of activating the current Transformer block. If not activated, the tokens from its preceding block won't pass through it. Specifically, let's consider the $i$-th layer ($i > n_f$). We denote the total number of tokens by $\mathcal{K}$, the embedding dimension of each token by $d$, and all the tokens output by the $(i-1)$-th layer by $\mathbf{t}_{1:\mathcal{K}}^{i-1}(Z,X) \in \mathbb{R}^{\mathcal{K}\times d}$. The slice of all tokens generated by the $(i-1)$-th Transformer block is expressed as $\mathbf{e}_1^{\mathrm{T}}\mathbf{t}_{1:\mathcal{K}}^{i-1}(Z,X) := \mathbf{r}^{i-1} \in \mathbb{R}^{\mathcal{K}}$, where $\mathbf{e}_1^{\mathrm{T}} = [1,0,...,0] \in \mathbb{R}^{\mathcal{K}}$ is a standard unit vector in $\mathbb{R}^{\mathcal{K}}$, the linear layer is denoted by $\mathfrak{L}^i$. Formally, the Activation Module (AM) at layer $i$ is defined by

$$p^i = \sigma(\mathfrak{L}^i(\mathbf{r}^{i-1})), \tag{1}$$

where $p^i \in [0,1]$ represents the activation probability of the $i$-th Transformer block, $\sigma(x) = 1/(1+e^{-x})$ indicates the sigmoid function. If $p^i > \beta$, where $\beta \in (0.5,1)$ is the activation probability threshold, the transformer block at layer $i$ will be activated; otherwise, it is deactivated and the output tokens from the $(i-1)$-th layer will be fed into the $(i+1)$-th block directly. Let $\mathcal{N}$ denote the total number of transformer blocks in the given ViT. Theoretically, deactivating all $\mathcal{N}$ blocks simultaneously would result in no correlation being computed between the template and search image. To avoid such unfavorable conditions, the first $n_f$ layers are mandated to remain activated. This strategy helps alleviate computational burdens associated with

AM, as these initial layers are typically essential, providing foundational information on which high-level and more abstract features and representations can be built. Another extreme situation is that, for whatever input, all Transformer blocks are activated so that the model could reduce classification and regression losses more easily and quickly as larger models exhibit more powerful fitting capabilities. To cope with this, we propose a block sparsity loss $\mathcal{L}_{spar}$ that penalizes higher mean probability of all adaptive layers so that, on average, many blocks are deactivated to enhance efficiency. The block sparsity loss is formulated as follows,

$$\mathcal{L}_{spar} = |\frac{1}{\mathcal{N} - n_f} \sum_{i=n_f+1}^{\mathcal{N}} p_i - \zeta|, \tag{2}$$

where $\zeta \in [0, 1]$ is a constant used in conjunction with $\beta$ to regulate block sparsity. We refer to $\zeta$ as the block sparsity constant. In general, for a given $\beta$, a smaller value of $\zeta$ results in a sparser model. Note that, if $\zeta = 0$, $p_i$ can be considered as the weight of the block $i$ and the sparsity loss is proportional to the $l_1$ norm of the vector consisting of these weights, thus a convex-relaxed sparsity regularization penalty commonly used in the area of statisitcal learning theory. $\zeta$ can be considered as a hyperparameter to facilitate finer adjustments.

### 3.3. View-Invariant Representations (VIR) via Mutual Information Maximization

To begin, we outline the idea of mutual information (MI) and establish the relevant notations. Let $\mathbf{a} \in \mathcal{A}$ and $\mathbf{b} \in \mathcal{B}$ represent two random variables. The MI between $\mathbf{a}$ and $\mathbf{b}$ is indicated by $\mathrm{I}(\mathbf{a}, \mathbf{b})$ and is expressed as follows:

$$\mathbb{E}_{p(\mathbf{a},\mathbf{b})}\left[log\frac{p(\mathbf{a},\mathbf{b})}{p(\mathbf{a})p(\mathbf{b})}\right] = \mathrm{D}_{KL}(p(\mathbf{a},\mathbf{b})||p(\mathbf{a})p(\mathbf{b})), \tag{3}$$

where $p(\mathbf{a}, \mathbf{b})$ represents the joint probability distribution, while $p(\mathbf{a})$ and $p(\mathbf{b})$ are the marginal distributions. The symbol $\mathrm{D}_{KL}$ denotes the Kullback–Leibler divergence (KLD) (MacKay, 2004). In practical scenarios, estimating Mutual Information (MI) is challenging, as we typically have access only to samples and not the underlying distributions (Poole et al., 2019). Many estimators were thus proposed to approximate the MI between variables based on observed samples. In this work, we leverage the MI estimator Deep InfoMax (R.D. & et al., 2019), which relies on Jensen-Shannon divergence (JSD) rather than KLD as the basis for the MI estimation. The selection of this method is driven by its established stability and effectiveness. The Jensen-Shannon MI estimator, represented by $\hat{\mathrm{I}}^{(JSD)}_{\mathbf{\Theta}}(\mathbf{a}, \mathbf{b})$, is defined by:

$$\mathbb{E}_{p(\mathbf{a},\mathbf{b})}[-\alpha(-T_{\mathbf{\Theta}}(\mathbf{a},\mathbf{b}))] - \mathbb{E}_{p(\mathbf{a})p(\mathbf{b})}[\alpha(T_{\mathbf{\Theta}}(\mathbf{a},\mathbf{b}))], \tag{4}$$

where $T_{\mathbf{\Theta}} : X \times Y \to \mathbb{R}$ is a neural network parameterized by $\mathbf{\Theta}$, and $\alpha(z) = log(1 + e^z)$ represents the softplus function.

In this study, our approach involves learning view-invariant feature representations by maximizing the MI with the above Jensen-Shannon MI estimator between the feature representations of two different views of the target. Let $\mathbf{t}^{\infty}_{1:\mathcal{K}}(Z, X) = \mathbf{t}^{\infty}_{\mathcal{K}_Z \cup \mathcal{K}_X}(Z, X), \mathcal{K}_Z, \cup \mathcal{K}_X = [1, \mathcal{K}]$, denote the final output tokens of the ViT, where $\mathbf{t}^{\infty}_{\mathcal{K}_Z}$ and $\mathbf{t}^{\infty}_{\mathcal{K}_X}$ represent the tokens corresponding to the template and the search image, respectively. Given the ground truth localization of the target, denoted by $Z'$ in the search image, we are able to obtain the tokens corresponding to $Z'$ with linear interpolation, which is denoted by $\mathbf{t}^{\infty}_{\mathcal{K}_{Z'}}(Z, X) \subset \mathbf{t}^{\infty}_{\mathcal{K}_X}(Z, X)$. The proposed loss $\mathcal{L}_{vir}$ for learning view-invariant feature representations is defined as follows,

$$\mathcal{L}_{vir} = -\hat{\mathrm{I}}^{(JSD)}_{\mathbf{\Theta}}(\mathbf{t}^{\infty}_{\mathcal{K}_{Z'}}(Z, X), \mathbf{t}^{\infty}_{\mathcal{K}_Z}(Z, X)). \tag{5}$$

As $\mathcal{L}_{vir}$ is exclusively computed during the training phase, our approach imposes no additional computational cost during the inference phase. Moreover, the proposed view-invariant representation learning is ViT-agnostic, which can be easily applied to other tracking frameworks.

### 3.4. Prediction Head and Training Loss

Following the corner detection head in (Cui et al., 2022; Ye et al., 2022a), we use a prediction head $\mathcal{H}$ comprising multiple Conv-BN-ReLU layers to directly estimate the target's bounding box. The output tokens associated with the search image are first turned into a 2D spatial feature map before being fed into the prediction head. The head ouputs a local offset $\mathbf{o} \in [0, 1]^{2\times H_x/P\times W_x/P}$, a normalized bounding box size $\mathbf{s} \in [0, 1]^{2\times H_x/P\times W_x/P}$, and a target classification score $\mathbf{p} \in [0, 1]^{H_x/P\times W_x/P}$ as prediction outcomes. The preliminary estimation of the target position relies on identifying the location with the highest classification score, i.e., $(x_c, y_c) = \mathrm{argmax}_{(x,y)}\mathbf{p}(x, y)$. Subsequently, the final target bounding box is estimated by

$$\{(x_t, y_t); (w, h)\} = \{(x_c, y_c) + \mathbf{o}(x_c, y_c); \mathbf{s}(x_c, y_c)\}. \tag{6}$$

As to training loss, we adopt the weighted focal loss (Law & Deng, 2018) for classification, a combination of $L_1$ loss and GIoU loss (Rezatofighi et al., 2019) for bounding box regression. Finally, the total loss function is given by

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda_{iou}\mathcal{L}_{iou} + \lambda_{L_1}\mathcal{L}_{L_1} + \gamma\mathcal{L}_{spar} + \kappa\mathcal{L}_{vir}, \tag{7}$$

where the constants $\lambda_{iou}$ = 2 and $\lambda_{L_1}$ = 5 are set as in (Cui et al., 2022; Ye et al., 2022a), $\gamma$ is set to 50 , $\kappa$ is set to 0.0001. Our framework undergoes end-to-end training using the overall loss $\mathcal{L}_{overall}$ after loading the pretrained weights of the ViT trained with ImageNet (Russakovsky et al., 2014).

## 4. Experiments

In this section, a comprehensive evaluation of our method is presented based on five UAV tracking benchmarks, i.e.,

| | Tracker | Source | DTB70 | | UAVDT | | VisDrone2018 | | UAV123 | | UAV123@10fps | | Avg. | | Avg.FPS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Prec. | Succ. | Prec. | Succ. | Prec. | Succ. | Prec. | Succ. | Prec. | Succ. | Prec. | Succ. | GPU | CPU |
| ViT-based | Ours | AVTrack-ViT | 81.3 | 63.3 | 79.9 | 57.7 | **86.4** | **65.9** | 84.0 | **66.2** | **83.2** | **65.7** | **82.9** | **63.8** | 250.2 | 59.7 |
| | | AVTrack-EVA | **82.6** | **64.0** | 78.8 | 57.2 | 83.4 | 62.5 | 83.0 | 64.7 | 81.2 | 63.5 | 81.8 | 62.3 | **283.7** | 62.8 |
| | | AVTrack-DeiT | **84.3** | **65.0** | **82.1** | 58.7 | **86.0** | **65.3** | **84.8** | **66.8** | **83.2** | **65.8** | **84.1** | **64.3** | **256.8** | 59.5 |
| | Aba-ViTrack(Li et al., 2023) | ICCV 23 | **85.9** | **66.4** | **83.4** | **59.9** | **86.1** | **65.3** | **86.4** | **66.4** | **85.0** | 65.5 | **85.3** | **64.7** | 181.5 | 50.3 |
| | LiteTrack (Wei et al., 2023a) | arXiv'23 | 82.5 | 63.9 | 81.6 | **59.3** | 79.7 | 61.4 | **84.2** | 65.9 | 83.1 | 65.0 | 82.2 | 63.1 | 141.6 | - |
| | SMAT(Gopal & Amer, 2024) | WACV 24 | 81.9 | 63.8 | 80.8 | 58.7 | 82.5 | **63.4** | 81.8 | 64.6 | 80.4 | 63.5 | 81.5 | 62.8 | 124.2 | - |
| CNN-based | HiFT(Cao et al., 2021) | 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 | 160.3 | - |
| | TCTrack(Cao et al., 2022) | 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 | 139.6 | - |
| | UDAT(Ye et al., 2022b) | CVPR 22 | 80.6 | 61.8 | 80.1 | **59.2** | 81.6 | 61.9 | 76.1 | 59.0 | 77.8 | 58.5 | 79.2 | 60.1 | 33.7 | - |
| | SGDViT(Yao et al., 2023) | ICRA 23 | 78.5 | 60.4 | 65.7 | 48.0 | 72.1 | 52.1 | 75.4 | 57.5 | **86.3** | **66.1** | 75.6 | 56.8 | 110.5 | - |
| | ABDNet(Zuo et al., 2023) | RAL 23 | 76.8 | 59.6 | 75.5 | 55.3 | 75.0 | 57.2 | 79.3 | 60.7 | 77.3 | 59.1 | 76.7 | 59.1 | 130.2 | - |
| | DRCI(Zeng et al., 2023) | ICME 23 | 81.4 | 61.8 | **84.0** | 59.0 | 83.4 | 60.0 | 76.7 | 59.7 | 73.6 | 55.2 | 79.8 | 59.1 | **281.3** | 62.4 |
| DCF-based | KCF(Henriques et al., 2015) | 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** |
| | fDSST(Danelljan et al., 2017) | 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** |
| | ECO_HC(Danelljan et al., 2017) | 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** |
| | MCCT_H(Wang et al., 2018) | 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 |
| | AutoTrack(Li et al., 2020b) | 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(Li et al., 2022a) | PR 20 | 72.6 | 50.5 | 77.3 | 49.4 | 83.4 | 60.0 | 70.2 | 47.7 | 69.4 | 48.6 | 74.6 | 81.2 | - | 35.6 |

*Table 1.* Comparison of precision (Prec.), success rate (Succ.), and speed (FPS) between AVTrack and lightweight trackers on DTB70, UAVDT, VisDrone2018, UAV123, and UAV123@10fps. **Red**, **blue**, and **green** signify the first, second, and third places. Please note that the percent symbol (%) is excluded for Prec. and Succ. values.

UAV123 (Mueller et al., 2016), UAV123@10fps (Mueller et al., 2016), VisDrone2018 (Zhu et al., 2018), UAVDT (Du et al., 2018), and DTB70 (Li & Yeung, 2017). Our evaluation is performed on a PC that was equipped with an i9-10850K processor (3.6GHz), 16GB of RAM, and an NVIDIA TitanX GPU. We assess our approach against a total of 13 state-of-the-art (SOTA) lightweight trackers (see Table 1) as well as 14 SOTA deep trackers developed specially for generic visual tracking (refer to Table 2).

### 4.1. Implementation Details

**Model.** We adopt different ViTs as backbones, including ViT-tiny (Dosovitskiy et al., 2021), DeiT-tiny (Touvron et al., 2021), and EVA-tiny (Fang et al., 2023), to build three trackers for evaluation, i.e., AVTrack-ViT, AVTrack-DeiT, and AVTrack-EVA, respectively. The head of AVTrack consists of a stack of four Conv-BN-Relu layers. The sizes of the search region and template are set to $256 \times 256$ and $128 \times 128$, respectively.

**Training.** We employ the training splits of multiple datasets for training, including GOT-10k (Huang et al., 2021), LaSOT (Fan et al., 2019), COCO (Lin et al., 2014), and TrackingNet (Muller et al., 2018). It is noteworthy that all three trackers share the same train pipeline for consistency and comparability. The batch size is uniformly fixed at 32. We utilize the AdamW optimizer with a weight decay of $10^{-4}$, and $4 \times 10^{-5}$ is used as the initial learning rate. The total number of training epochs is uniformly fixed at 300, with 60,000 image pairings processed every epoch, and the learning rate drops by a factor of 10 after 240 epochs.

**Inference.** In accordance with conventional practices (Zhang et al., 2020), Hanning window penalties are applied during inference to incorporate positional prior in tracking. Specifically, we execute a multiplication of the classification map by a Hanning window of the same size. Subsequently, the box exhibiting the highest score is designated as the result of the tracking process.

### 4.2. Comparison with Lightweight Trackers

The evaluation results of our trackers and the rival lightweight trackers are shown in Table 1. As can be seen, our AVTrack demonstrate superior performance among all these trackers in terms of average (Avg.) precision (Prec.), success rate (Succ.) and speeds. On average, RACF (Li et al., 2022a) demonstrated the highest Prec. (74.6%) and Succ. (51.2%) among DCF-based trackers, DRCI (Zeng et al., 2023) achieves the highest Prec. at 79.8%, while UDAT (Ye et al., 2022b) attains the highest Succ. of 60.1% among CNN-based trackers. However, the Prec. and Succ. of all ViT-based trackers are greater than 81.0% and 62.0%, respectively, clearly surpassing DCF- and CNN- based approaches. When considering GPU speed, AVTrack-EVA stands out with the highest speed of 283.7 FPS. Following closely are DRCI and AVTrack-DeiT, achieving the second and third positions with speeds of 281.3 FPS and 256.8 FPS, respectively. However, despite having a GPU speed comparable to AVTrack-DeiT, DRCI shows significantly lower Avg. Prec. and Succ. compared to AVTrack-DeiT. Regarding CPU speed, all of our trackers demonstrate real-time performance on a single CPU[1], even faster than some DCF-based trackers such as AutoTrack and RACF. Despite Aba-ViTrack achieving the top Avg. performance with 85.3% Avg. Prec. and 64.7% Avg. Succ., AVTrack-DeiT claims the second spot with only slight gaps of 1.2% and 0.4%, respectively. Notably, AVTrack-ViT outperforms

[1]Note that the real-time performance discussed in this paper may only apply to platforms similar to or more advanced than ours.

| Method | Prec. | Succ. | FPS | Tracker | Prec. | Succ. | FPS | Tracker | Prec. | Succ. | FPS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **AVTrack-DeiT (Ours)** | **86.0** | **65.3** | **220.0** | SLT-TrDiMP(Kim et al., 2022) | 85.1 | 63.6 | 27.3 | KeepTrack(Mayer et al., 2021) | 84.0 | 63.5 | 18.7 |
| ROMTrack(Cai et al., 2023) | **86.3** | **66.7** | 51.1 | OSTrack(Ye et al., 2022a) | 84.2 | 64.8 | **66.0** | TransT(Chen et al., 2021a) | **85.9** | 65.2 | 51.7 |
| SeqTrack(Chen et al., 2023) | 85.3 | **65.8** | 11.0 | SLT-TransT(Kim et al., 2022) | 85.6 | **65.3** | 29.5 | TrSiam(Wang et al., 2021) | 84.7 | 64.0 | 32.8 |
| MAT(Zhao et al., 2023) | 81.6 | 62.2 | **71.2** | ToMP(Mayer et al., 2022) | 84.1 | 64.4 | 21.6 | PrDiMP50(Danelljan et al., 2020) | 79.4 | 59.7 | 41.3 |
| SparseTT(Fu et al., 2022) | 81.4 | 62.1 | 28.3 | AutoMatch(Zhang et al., 2021) | 78.1 | 59.6 | 62.1 | SiamRPN++(Li et al., 2019) | 79.1 | 60.0 | 55.1 |

*Table 2.* Precision (Prec.), success (Succ.), and GPU speed comparison between our AVTrack-Deit and DL-based tracker on VisDrone2018.

Aba-ViTrack in both Prec. and Succ. on VisDrone2018, while AVTrack-DeiT surpasses Aba-ViTrack in Succ. on UAV123. Moreover, all our trackers demonstrate higher speeds compared to Aba-ViTrack. Remarkably, AVTrack-DeiT achieves over 1.4 times the GPU speed and 1.2 times the CPU speed of Aba-ViTrack, demonstrating a superior balance between tracking precision and efficiency, underscoring the advantages of our method and substantiate its SOTA performance in UAV tracking.

### 4.3. Comparison with Deep Trackers

The proposed AVTrack-DeiT is also compared with 14 deep trackers, as illustrated in Table 2. The table presents the Prec., Succ., and GPU speed of the competing trackers on VisDrone2018. Notably, our AVTrack-DeiT stands out by achieving the second-ranked Prec., the third-ranked Succ., and the fastest GPU speed, showcasing its competitiveness in both accuracy and speed. Also note that the gaps of Prec. and Succ. of our method to the first place are slight, only 0.3% and 1.4%, respectively. While several deep trackers, including ROMTrack, SLT-TransT, and TansT, achieve comparable accuracy to AVTrack-DeiT, their GPU speeds are significantly slower. For instance, our method is 3, 6, and 3 times faster than ROMTrack, SLT-TransT, and TansT, respectively.

### 4.4. Attribute-Based Evaluation

To evaluate the robustness of the proposed method against view variances of targets, we compare AVTrack-DeiT with 20 SOTA trackers on the viewpoint change subset of the VisDrone2018. Note that we also assessed AVTrack-DeiT without applying the proposed method for learning View-Invariant Representations (VIR), denoted as AVTrack-DeiT* for reference. The precision plot is depicted in Fig. 3, and additional attribute-based evaluation results can be found in the supplemental materials. As observed, AVTrack-DeiT achieves the third-highest precision, with a Prec. of 85.4%. Notably, the incorporation of the proposed components results in a substantial improvement over AVTrack-DeiT* by 3.1% in precision, underscoring the effectiveness of the proposed method.

### 4.5. Ablation Study

**Impact of Activation Module (AM) and View-Invariant Representations (VIR).** To support the effectiveness of

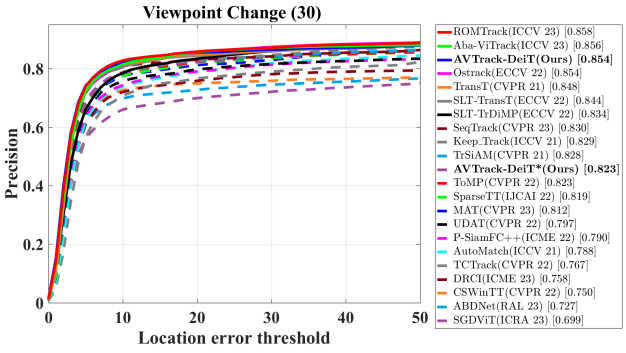


*Figure 3.* Attribute-based comparison on the viewpoint change subset of VisDrone2018. Note that **AVTrack-DeiT*** refers to **AVTrack-DeiT** without utilizing the proposed VIR component.

| Method | VIR | AM | Prec. | Succ. | FPS |
|---|---|---|---|---|---|
| AVTrack-ViT | | | 83.0 | 62.7 | 188.3 |
| | ✓ | | **87.1**$_{\uparrow 4.1}$ | **66.4**$_{\uparrow 3.7}$ | - |
| | ✓ | ✓ | 86.4$_{\uparrow 3.4}$ | 65.9$_{\uparrow 3.2}$ | **238.6**$_{\uparrow 26\%}$ |
| AVTrack-EVA | | | 79.7 | 60.7 | 235.6 |
| | ✓ | | **84.3**$_{\uparrow 4.6}$ | **63.2**$_{\uparrow 2.5}$ | - |
| | ✓ | ✓ | 83.4$_{\uparrow 3.7}$ | 62.5$_{\uparrow 1.8}$ | **285.6**$_{\uparrow 21\%}$ |
| AVTrack-DeiT | | | 82.3 | 63.2 | 192.7 |
| | ✓ | | **86.7**$_{\uparrow 4.4}$ | **65.9**$_{\uparrow 2.7}$ | - |
| | ✓ | ✓ | 86.0$_{\uparrow 3.7}$ | 65.3$_{\uparrow 2.1}$ | **220.0**$_{\uparrow 14\%}$ |

*Table 3.* Impact of AM and VIR on the performance of the baseline trackers on VisDrone2018.

the proposed AM and VIR, Table 3 presents the evaluation results on VisDrone2018 by gradually incorporating these components into the baselines. To eliminate any potential nuances arising from randomness, we only present the speed of the baseline since the GPU speeds of the baseline and its VIR-enhanced version are theoretically equal. As can be seen, the incorporation of VIR significantly enhances both Prec. and Succ. for all baseline trackers. Specifically, the Prec. increases for AVTrack-ViT, AVTrack-EVA, and AVTrack-DeiT are 4.1%, 4.6%, and 4.4%, respectively, while the Succ. increases are 3.7%, 2.5%, and 2.7%, respectively. These significant enhancements highlight the effectiveness of VIR in improving tracking precision. The further integration of AM leads to consistent enhancements in GPU speeds, with only slight reductions in Prec. and Succ. Specifically, all baseline trackers undergo GPU speed enhancements of more than 14.0% with AVTrack-ViT demonstrating an outstanding improvement of 26.0%, affirming the effectiveness of AM in optimizing tracking efficiency

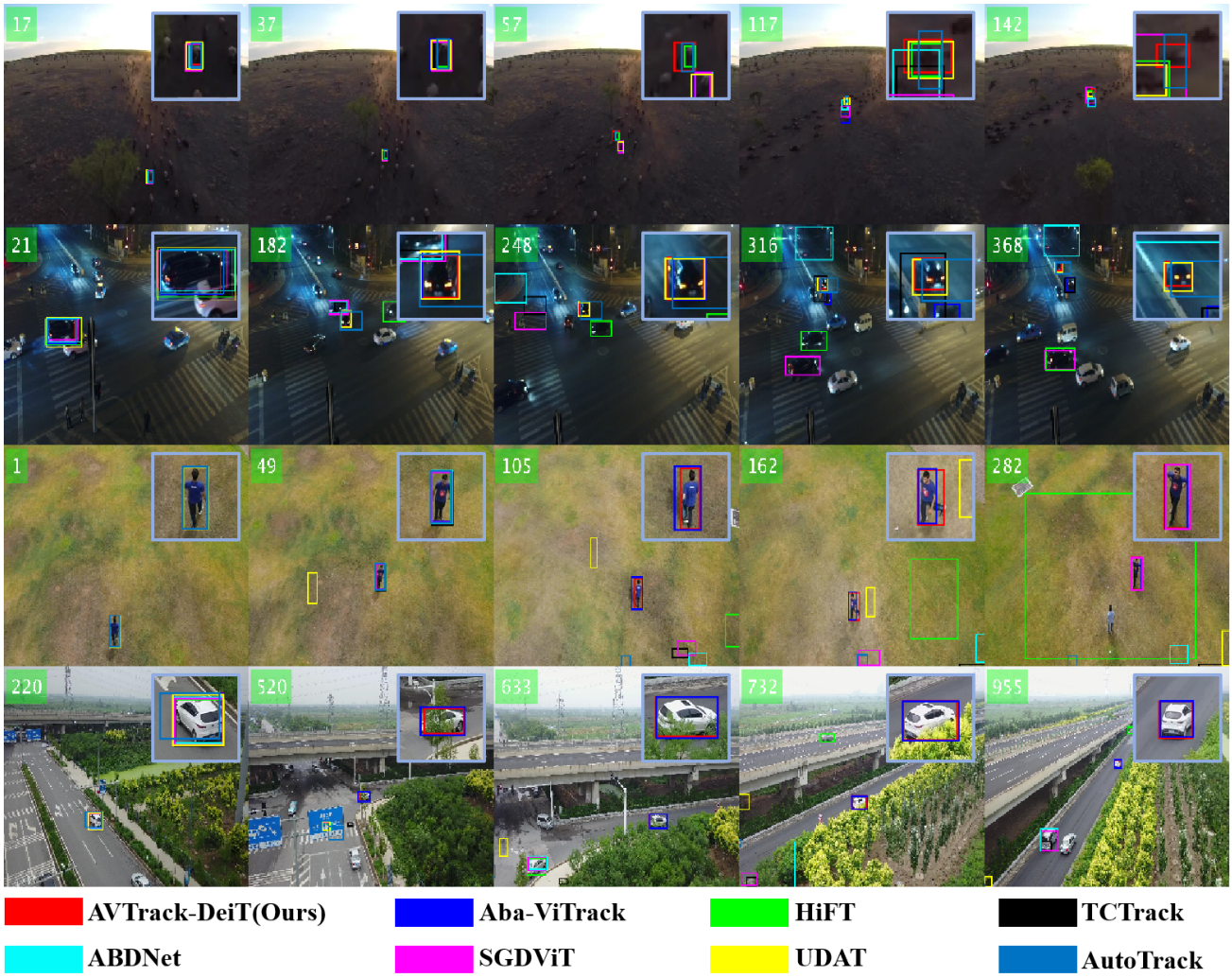


*Figure 4.* Qualitative evaluation on 4 video sequences from, respectively, DTB70, UAVDT, UAV123@10fps, and VisDrone2018 (i.e. Animal2, S301, person10, and uav0000180_00050_s).

| Method | VIR | AM | Prec. | Succ. | FPS |
|---|---|---|---|---|---|
| | | | 77.7 | 59.5 | 77.5 |
| ARTrack(Wei et al., 2023b) | ✓ | | **79.9**$_{\uparrow 2.2}$ | **60.8**$_{\uparrow 1.3}$ | - |
| | ✓ | ✓ | 79.3$_{\uparrow 1.6}$ | 60.4$_{\uparrow 0.9}$ | **94.5**$_{\uparrow 22\%}$ |
| | | | 81.5 | 62.7 | 177.4 |
| DropTrack(Wu et al., 2023) | ✓ | | **83.1**$_{\uparrow 1.6}$ | **64.0**$_{\uparrow 1.3}$ | - |
| | ✓ | ✓ | 82.7$_{\uparrow 1.2}$ | 63.6$_{\uparrow 0.9}$ | **214.5**$_{\uparrow 21\%}$ |
| | | | 82.7 | 63.4 | 198.5 |
| GRM(Gao et al., 2023) | ✓ | | **84.1**$_{\uparrow 1.4}$ | **64.5**$_{\uparrow 1.1}$ | - |
| | ✓ | ✓ | 83.7$_{\uparrow 1.0}$ | 64.1$_{\uparrow 0.7}$ | **234.7**$_{\uparrow 18\%}$ |

*Table 4.* Evaluation of the generalizability of our VIR and AM by applying them to three SOTA trackers. Note that we substitute their backbones to ViT-Tiny to save training time. The evaluation is performed on VisDrone2018.

with only minor impact on tracking performance.

**Application to SOTA trackers.** To demonstrate the generalizability of our approach, we incorporate the proposed VIR and AM into three state-of-the-art (SOTA) trackers: ARTrack (Wei et al., 2023b), GRM (Gao et al., 2023), and DropTrack (Wu et al., 2023). Note that we substitute their original backbones with the tiny ViT, i.e., ViT-Tiny (Dosovitskiy et al., 2021), to save training time. The evaluation results on VisDrone2018 are presented in Table 4. Similar to the preceding part, we provide only the speed of the baseline to eliminate potential nuances arising from randomness. As can be seen, incorporating VIR leads to substantial improvements in both Prec. and Succ. for all baseline trackers. Specifically, the Prec. increases for ARTrack, DropTrack, and GRM by 2.2%, 1.6%, and 1.4%, respectively, while the Succ. increases are 1.3%, 1.3%, and 1.1%, respectively. Further integration of AM consistently results in enhanced GPU speeds, with only slight decreases in Prec. and Succ. Specifically, the GPU speeds of all baseline trackers show

improvements of more than 18.0%, while the tracking performance experiences only marginal drops of less than 0.6%. These outcomes well confirm the generalizability of our method.

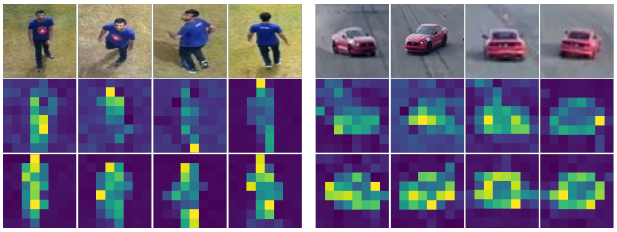

*Figure 5.* For each group, we display the target images from various viewpoints (top), followed by a feature map generated by AVTrack-DeiT without the proposed VIR (middle), and with the proposed VIR (bottom).

### 4.6. Qualitative Results

Fig. 4 showcases the qualitative tracking results generated by AVTrack-DeiT and 7 SOTA trackers. Remarkably, our tracker stands out as the only one successfully tracking all the targets in all these challenging examples, including scenarios with background clusters (i.e., Animal2), scale variations (i.e., S0301 and uav0000180_00050_s), and pose variations (e.g., across all sequences). The superior performance and visual appeal of our method in these challenging scenarios provide additional evidence of the effectiveness of the proposed approach for UAV tracking.

Fig. 5 illustrates the feature maps for two examples from UAV123 (Mueller et al., 2016), as generated by AVTrack-DeiT before and after integrating the proposed component VIR. In each group, the top row showcases different frames from the same video sequence, each offering a unique perspective that enables a comparison of the visual representations and showcases the impact of the proposed VIR component. The feature maps generated by AVTrack-DeiT exhibit more consistency than the baseline in the presense of viewpoint changes, further demonstrating the effectiveness of our method in learning view-invariant robust feature representations using ViTs.

## 5. Conclusions

In this paper, our focus was on investigating the effectiveness of a unified framework for real-time UAV tracking using efficient Vision Transformers (ViTs). To achieve this, we streamlined the framework by implementing an adaptive computation paradigm that selectively activates Transformer blocks. Furthermore, to tackle the challenges presented by significant changes in viewing angles commonly encountered in UAV tracking, we utilized mutual information maximization to learn view-invariant representations. Thanks to its simplicity, our method can be seamlessly integrated or adapted into other ViT-based trackers. Exhaustive experiments across five UAV tracking benchmarks validate the effectiveness of our method and show that our AVTrack-DeiT achieves SOTA performance in UAV tracking.

## Acknowledgements

This research was partly funded by the Guangxi Natural Science Foundation (Grant No. 2024GXNSFAA010484), the Guangxi Science and Technology Base and Talent Special Project (Grant No. Guike AD22035127), the Guangxi Key Technologies R&D Program (Grant No. Guike AB23049001, Guike AB23026004) and the National Natural Science Foundation of China (Grant No. 62262011).

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References

Bertinetto, L., Valmadre, J., and et al. Staple: Complementary learners for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Bracci, S., Caramazza, A., and Peelen, M. V. View-invariant representation of hand postures in the human lateral occipitotemporal cortex. *NeuroImage*, 2018.

Cai, Y., Liu, J., Tang, J., and Wu, G. Robust object modeling for visual tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Cao, Z., Fu, C., Ye, J., Li, B., and Li, Y. Hift: Hierarchical feature transformer for aerial tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Cao, Z., Huang, Z., Pan, L., Zhang, S., Liu, Z., and Fu, C. Tctrack: Temporal contexts for aerial tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chen, B., Li, P., Bai, L., Qiao, L., Shen, Q., Li, B., Gan, W., Wu, W., and Ouyang, W. Backbone is All Your Need: A Simplified Architecture for Visual Object Tracking. In *European Conference on Computer Vision (ECCV)*, 2022.

Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., and Lu, H. Transformer tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021a.

Chen, X., Peng, H., Wang, D., Lu, H., and Hu, H. Seqtrack: Sequence to sequence learning for visual object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., and Liu, Z. Mobile-former: Bridging mobilenet and transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Cui, Y., Jiang, C., Wang, L., and Wu, G. Mixformer: End-to-end tracking with iterative mixed attention. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Danelljan, M., Khan, F. S., Felsberg, M., and van de Weijer, J. Adaptive color attributes for real-time visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Danelljan, M., Bhat, G., Shahbaz Khan, F., and Felsberg, M. Eco: Efficient convolution operators for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Danelljan, M., Hager, G., Khan, F. S., and Felsberg, M. Discriminative scale space tracking. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2017.

Danelljan, M., Gool, L. V., and Timofte, R. Probabilistic regression for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Du, D., Qi, Y., Yu, H., Yang, Y.-F., Duan, K., Li, G., Zhang, W., Huang, Q., and Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In *European Conference on Computer Vision (ECCV)*, 2018.

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., and Ling, H. Lasot: A high-quality benchmark for large-scale single object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., and Cao, Y. Eva: Exploring the limits of masked visual representation learning at scale. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Feng, C., Jie, Z., Zhong, Y., Chu, X., and Ma, L. Aedet: Azimuth-invariant multi-view 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Fu, Z., Fu, Z., Liu, Q., Cai, W., and Wang, Y. Sparsett: Visual tracking with sparse transformers. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

Gao, L., Ji, Y., Gedamu, K., Zhu, X., Xu, X., and Shen, H. T. View-invariant human action recognition via view transformation network (vtn). *IEEE Transactions on Multimedia*, 2022.

Gao, S., Zhou, C., and Zhang, J. Generalized relation modeling for transformer tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Gopal, G. Y. and Amer, M. A. Separable self and mixed attention transformers for efficient object tracking. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., and Shen, C. Graph attention tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Henriques, J. F., Caseiro, R., and et al. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2015.

Huang, L., Zhao, X., and Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2021.

Huang, Z., Fu, C., and et al. Learning aberrance repressed correlation filters for real-time uav tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Ji, X. and Liu, H. Advances in view-invariant human motion analysis: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2010.

Kim, M., Lee, S., Ok, J., Han, B., and Cho, M. Towards sequence-level training for visual tracking. In *European Conference on Computer Vision (ECCV)*, 2022.

Kumie, G. A., Habtie, M. A., Ayall, T. A., Zhou, C., Liu, H., Seid, A. M., and Erbad, A. Dual-attention network for view-invariant action recognition. *Complex & Intelligent Systems*, 2024.

Law, H. and Deng, J. Cornernet: Detecting objects as paired keypoints. In *European conference on computer vision (ECCV)*, 2018.

Li, B., Wu, W., and et al. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Li, C., Min, X., Sun, S., Lin, W., and Tang, Z. Deepgait: A learning deep convolutional representation for view-invariant gait recognition using joint bayesian. *Applied Sciences*, 2017.

Li, S. and Yeung, D. Y. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Li, S., Jiang, Q., Zhao, Q., Lu, L., and Feng, Z. Asymmetric discriminative correlation filters for visual tracking. *Frontiers of Information Technology & Electronic Engineering*, 21(10):1467–1484, 2020a.

Li, S., Liu, Y., Zhao, Q., and Feng, Z. Learning residue-aware correlation filters and refining scale estimates with the grabcut for real-time uav tracking. In *2021 International Conference on 3D Vision (3DV)*, pp. 1238–1248. IEEE, 2021a.

Li, S., Zhao, Q., Feng, Z., and Lu, L. Equivalence of correlation filter and convolution filter in visual tracking. In *Image and Graphics*, pp. 623–634, Cham, 2021b. Springer International Publishing.

Li, S., Liu, Y., Zhao, Q., and Feng, Z. Learning residue-aware correlation filters and refining scale for real-time uav tracking. *Pattern Recognition (PR)*, 2022a.

Li, S., Yang, Y., Zeng, D., and Wang, X. Adaptive and background-aware vision transformer for real-time uav tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Li, Y. and Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision (ECCV)*, 2015.

Li, Y., Fu, C., and et al. Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020b.

Li, Y., Yuan, G., Wen, Y., Hu, E., Evangelidis, G., Tulyakov, S., Wang, Y., and Ren, J. Efficientformer: Vision transformers at mobilenet speed. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

Liu, M., Wang, Y., Sun, Q., and Li, S. Global filter pruning with self-attention for real-time uav tracking. In *British Machine Vision Conference (BMVC)*, 2022a.

Liu, Z., Feng, R., Chen, H., Wu, S., Gao, Y., Gao, Y., and Wang, X. Temporal feature alignment and mutual information maximization for video-based human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

M., D. and et al. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ma, S., Liu, Y., Zeng, D., Liao, Y., Xu, X., and Li, S. Learning disentangled representation in pruning for real-time uav tracking. In *Asian Conference on Machine Learning (ACML)*, 2023.

MacKay, D. J. *Information theory, inference and learning algorithms*. Cambridge university press, 2004.

Mao, J., Yang, H., Li, A., Li, H., and Chen, Y. Tprune: Efficient transformer pruning for mobile devices. *ACM Transactions on Cyber-Physical Systems (TCPS)*, 2021.

Mayer, C., Danelljan, M., Paudel, D. P., and Van Gool, L. Learning target candidate association to keep track of what not to track. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Mayer, C., Danelljan, M., Bhat, G., Paul, M., Paudel, D. P., Yu, F., and Gool, L. V. Transforming model prediction for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Mueller, M., Smith, N., and Ghanem, B. A benchmark and simulator for uav tracking. In *European Conference on Computer Vision (ECCV)*, 2016.

Mueller, M., Smith, N., and Ghanem, B. Context-aware correlation filter tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., and Ghanem, B. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *European Conference on Computer Vision (ECCV)*, 2018.

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *International Conference on Machine Learning (ICML)*, 2019.

Rao, C., Yilmaz, A., and Shah, M. View-invariant representation and recognition of actions. *International journal of computer vision (IJCV)*, 2002.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

R.D., H. and et al. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019.

Rezatofighi, S. H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I. D., and Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 2014.

Shiraga, K., Makihara, Y., Muramatsu, D., Echigo, T., and Yagi, Y. Geinet: View-invariant gait recognition using a convolutional neural network. In *International conference on biometrics (ICB)*, 2016.

Steuer, R., Kurths, J., Daub, C. O., Weise, J., and Selbig, J. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 2002.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021.

Wang, N., gang Zhou, W., Tian, Q., Hong, R., Wang, M., and Li, H. Multi-cue correlation filters for robust visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Wang, N., Zhou, W., Wang, J., and Li, H. Transformer Meets Tracker: Exploiting Temporal Context for Robust Visual Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wang, X., Zeng, D., Zhao, Q., and Li, S. Rank-based filter pruning for real-time uav tracking. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2022.

Wang, X., Yang, X., Ye, H., and Li, S. Learning disentangled representation with mutual information maximization for real-time uav tracking. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2023.

Wei, Q., Zeng, B., Liu, J., He, L., and Zeng, G. Litetrack: Layer pruning with asynchronous feature extraction for lightweight and efficient visual tracking. *arXiv preprint arXiv:2309.09249*, 2023a.

Wei, X., Bai, Y., Zheng, Y., Shi, D., and Gong, Y. Autoregressive visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b.

Wu, Q., Yang, T., Liu, Z., Wu, B., Shan, Y., and Chan, A. B. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Wu, W., Zhong, P., and Li, S. Fisher pruning for real-time uav tracking. In *International Joint Conference on Neural Networks (IJCNN)*, 2022.

Xia, L., Chen, C.-C., and Aggarwal, J. K. View invariant human action recognition using histograms of 3d joints. In *IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW)*, 2012.

Xie, F., Wang, C., Wang, G., Yang, W., and Zeng, W. Learning tracking representations via dual-branch fully transformer networks. In *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021.

Xie, F., Wang, C., Wang, G., Cao, Y., Yang, W., and Zeng, W. Correlation-aware deep tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yang, X., Yan, J., Cheng, Y., and Zhang, Y. Learning deep generative clustering via mutual information maximization. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2022.

Yao, L., Fu, C., and et al. Sgdvit: Saliency-guided dynamic vision transformer for uav tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Ye, B., Chang, H., Ma, B., Shan, S., and Chen, X. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision (ECCV)*, 2022a.

Ye, J., Fu, C., Zheng, G., Paudel, D. P., and Chen, G. Unsupervised domain adaptation for nighttime aerial tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Yin, H., Vahdat, A., Alvarez, J. M., Mallya, A., Kautz, J., and Molchanov, P. A-vit: Adaptive tokens for efficient vision transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Zeng, D., Zou, M., Wang, X., and Li, S. Towards discriminative representations with contrastive instances for real-time uav tracking. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2023.

Zhang, J., Peng, H., Wu, K., Liu, M., Xiao, B., Fu, J., and Yuan, L. Minivit: Compressing vision transformers with weight multiplexing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Zhang, Z., Peng, H., Fu, J., Li, B., and Hu, W. Ocean: Object-aware anchor-free tracking. In *European Conference on Computer Vision (ECCV)*, 2020.

Zhang, Z., Liu, Y., Wang, X., Li, B., and Hu, W. Learn to match: Automatic matching network design for visual tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Zhang, Z., Wu, F., Qiu, Y., Liang, J., and Li, S. Tracking small and fast moving objects: A benchmark. In *Asian Conference on Computer Vision (ACCV)*, 2022b.

Zhao, H., Wang, D., and Lu, H. Representation learning for visual object tracking by masked appearance transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Zhong, P., Zeng, D., Wang, X., and Li, S. Efficiency and precision trade-offs in uav tracking with filter pruning and dynamic channel weighting. In *Fuzzy Systems and Data Mining (FSDM)*, 2022.

Zhong, P., Wu, W., Dai, X., Zhao, Q., and Li, S. Fisher pruning for developing real-time uav trackers. *Journal of Real-Time Image Processing*, 2023.

Zhou, Z., Pei, W., Li, X., Wang, H., Zheng, F., and He, Z. Saliency-associated object tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Zhu, P., Wen, L., and et al. Visdrone-sot2018: The vision meets drone single-object tracking challenge results. In *European Conference on Computer Vision (ECCV)*, 2018.

Zuo, H., Fu, C., Li, S., Lu, K., Li, Y., and Feng, C. Adversarial blur-deblur network for robust uav tracking. *IEEE Robotics and Automation Letters (RAL)*, 2023.

# Appendices

## A. Comparison with Deep Trackers

To adhere to the page length limitation in our main paper, we conducted a comparison solely between our approach and 14 state-of-the-art deep trackers on the VisDrone2018 (Zhu et al., 2018) dataset. Table A1 presents the evaluation of our AVTrack-DeiT's performance compared to these trackers on three additional datasets, namely DTB70 (Li & Yeung, 2017), UAVDT (Du et al., 2018), and UAV123@10fps (Mueller et al., 2016). The table displays the results for precision (Prec.) and success rate (Succ.), their respective averages (Avg.), and average FPS (Avg.FPS). The values for precision and success rate are presented in the form of (Prec, Succ). Notably, our AVTrack-DeiT stands out with its unparalleled GPU speed, maintaining high performance and showcasing a competitive edge in both accuracy and speed. Using the top three average performance trackers on four datasets, namely ToMP, KeepTrack, and ROMTrack, as examples, we can observe that while they may have slightly better Avg. Prec. and Succ. than our AVTrack-DeiT, their GPU speeds are significantly slower. Specifically, our method outperforms ToMP, KeepTrack, and ROMTrack by being 11, 12 and 5 times faster, respectively.

| Trcker | DTB70 | UAVDT | VisDrone2018 | UAV123@10fps | Avg. | Avg.FPS |
|---|---|---|---|---|---|---|
| **AVTrack-DeiT(Ours)** | (84.3, 65.0) | (82.1, 58.7) | (**86.0**, **65.3**) | (83.2, 65.8) | (83.9, 63.7) | **253.4** |
| ROMTrack(Cai et al., 2023) | (**87.1**, **67.4**) | (81.9, 61.6) | (**86.3**, **66.7**) | (85.0, 67.7) | (**85.1**, **65.9**) | 53.1 |
| MAT(Gao et al., 2023) | (83.2, 64.5) | (72.9, 54.6) | (81.6, 62.2) | (**89.3**, **68.6**) | (81.8, 62.5) | **72.3** |
| SeqTrack(Chen et al., 2023) | (**85.7**, 65.6) | (79.0, 59.8) | (85.3, **65.8**) | (**88.2**, **68.1**) | (84.6, 64.8) | 17.6 |
| SparseTT(Fu et al., 2022) | (82.3, **65.8**) | (82.8, **65.4**) | (81.4, 62.1) | (82.2, 64.9) | (82.2, 64.6) | 31.5 |
| OSTrack(Ye et al., 2022a) | (82.7, 65.1) | (**85.0**, **67.2**) | (84.2, 64.8) | (85.5, 66.1) | (84.4, **65.8**) | 65.8 |
| SLT-TranT(Kim et al., 2022) | (83.4, 65.6) | (82.9, 62.5) | (85.6, **65.3**) | (86.2, 67.4) | (84.5, 65.2) | 32.6 |
| ToMP(Mayer et al., 2022) | (**85.6**, **67.1**) | (**85.4**, 64.1) | (84.1, 64.4) | (87.5, 67.9) | (**85.7**, **65.8**) | 23.8 |
| AutoMatch(Zhang et al., 2021) | (82.5, 63.4) | (82.1, 60.8) | (78.1, 59.6) | (85.2, 65.7) | (81.9, 62.3) | 35.2 |
| KeepTrack(Mayer et al., 2021) | (83.6, 64.3) | (**83.8**, 60.5) | (84.0, 63.5) | (**89.7**, **68.2**) | (**85.2**, 64.1) | 20.3 |
| TransT(Chen et al., 2021a) | (83.6, **65.8**) | (82.6, **64.2**) | (**85.9**, 65.2) | (84.8, 66.5) | (84.2, **65.4**) | 55.0 |
| SAOT(Zhou et al., 2021) | (83.1, 64.6) | (82.1, 60.7) | (76.9, 59.1) | (85.2, 65.7) | (81.8, 62.5) | 35.2 |
| SiamGAT(Guo et al., 2020) | (75.1, 57.9) | (76.4, 58.9) | (78.3, 59.2) | (77.6, 59.7) | (76.9, 58.9) | **95.8** |
| PrDiMP50(Danelljan et al., 2020) | (76.4, 59.5) | (82.7, 60.1) | (79.4, 59.7) | (**87.9**, 67.5) | (81.6, 59.7) | 42.3 |
| SiamRPN++(Li et al., 2019) | (79.9, 61.4) | (82.2, 61.0) | (79.1, 60.0) | (78.4, 59.4) | (79.9, 60.5) | 57.6 |

*Table A1.* The comparison of the precision (Prec.), success rate (Succ.), and speed (FPS) of deep-based trackers on DTB70, UAVDT, VisDrone2018 and UAV123@10fps with AVTrack-DeiT. Note that precision and success rate are represented in the form of (Prec., Succ.), while the average GPU speed is presented as GPU fps. The top three results are displayed in **red**, **blue** and **green** fonts.

## B. Impact of Weighting the Loss for Learning View-Invariant Feature Representations

We utilize Table A2 to provide the comprehensive analysis of the performance, assessed across all five datasets, of the proposed loss ($\mathcal{L}_{vir}$) for learning view-invariant feature representations. The data presented in the table indicates that our tracker achieves the best performance when the value of $\kappa$ is set to $1 \times 10^{-4}$. Furthermore, we've noted that there are no discernible trends among the second and third-best performances, which are dispersed both above and below the value of $1 \times 10^{-4}$. When the $\kappa$ is set to 0.5, we observe an average maximal difference of 1.6% in Prec. and a maximal difference of 1.2% in Succ. These substantial margins clearly indicate that the selection of weight significantly influences the tracking performance. Specifically, when the loss is weighted appropriately, the tracker can attain optimal performance. Conversely, if the weight is not chosen properly, it can lead to unsatisfactory performance.

## C. Impact of Activation Module (AM) and View-Invariant Representations (VIR).

Table A3 provides a detailed analysis of the impact of activation module and view-invariant representations, evaluated across all five datasets. It is important to note that the GPU speed of both the baseline and its enhanced version by VIR are theoretically equal. In order to minimize any potential variations due to randomness, we only present the speed of the baseline. As shown in the table, when the VIR component is incorporated, the Avg. Prec. increases for AVTrack-ViT, AVTrack-EVA, and AVTrack-DeiT are 2.6%, 3.6%, and 2.5%, respectively, while the Avg. Succ. increases are 1.8%, 1.5%, and 1.6%, respectively. The integration of AM leads to consistent enhancements in GPU speeds, with only slight reductions

| $\kappa$ | DTB70 | UAVDT | VisDrone2018 | UAV123 | UAV123@10fps | Avg. |
|---|---|---|---|---|---|---|
| 0.5 | (82.3, 63.2) | (81.1, 57.9) | (84.2, 64.1) | (83.1, 65.6) | (82.1, 64.9) | (82.5, 63.1) |
| 0.6 | (82.8, 63.5) | (80.5, 57.4) | (84.7, 64.3) | (83.4, 65.8) | (82.6, 65.3) | (82.8, 63.3) |
| 0.7 | (83.5, 64.4) | (82.2, 58.8) | (85.4, 64.8) | (82.9, 65.4) | (81.8, 64.7) | (83.2, 63.6) |
| 0.8 | (83.0, 64.2) | (81.7, 58.5) | (83.9, 63.8) | (83.6, 65.9) | (**83.1**,**65.7**) | (83.1, 63.6) |
| 0.9 | (83.8, 64.6) | (**82.6**, **59.0**) | (85.0, 64.6) | (84.1, **66.3**) | (**83.8**, **66.2**) | (**83.9**, **64.1**) |
| 1.0 | (**84.3**, **65.0**) | (82.1, 58.7) | (**86.0**, **65.3**) | (**84.8**, **66.8**) | (**83.2**, **65.8**) | (**84.1**, **64.3**) |
| 1.1 | (82.9, 63.5) | (**82.4**, 58.9) | (**86.5**, **65.6**) | (**84.4**, **66.6**) | (82.5, 65.2) | (**83.7**, **64.0**) |
| 1.2 | (82.5, 63.3) | (**83.1**, **59.3**) | (85.1, 64.7) | (83.8, 66.0) | (83.0, 65.6) | (83.5, 63.9) |
| 1.3 | (**84.0**, **64.8**) | (81.8, 58.5) | (**85.6**, **65.0**) | (**84.2**, **66.3**) | (82.2, 64.9) | (83.6, 63.9) |
| 1.4 | (**84.1**, **64.9**) | (81.3, 58.2) | (84.8, 64.4) | (83.4, 64.7) | (82.6, 65.2) | (83.2, 63.5) |
| 1.5 | (83.4, 64.4) | (**82.4**, **59.1**) | (84.6, 64.3) | (83.1, 64.5) | (81.7, 64.7) | (83.0, 63.4) |

*Table A2.* By changing $\kappa$ from $0.5 \times 10^{-4}$ to $1.5 \times 10^{-4}$, an ablation research was conducted on loss for learning view-invariant feature representations ($\mathcal{L}_{vir}$) weighting for DTB70, UAVDT, VisDrone2018, UAV123, and UAV123@10fps. Please take note that $\times 10^{-4}$ has been omitted for clarity. The Prec. and Succ. are shown as (Prec., Succ.).

| Tracker | VIR | AM | DTB70 | UAVDT | VisDrone2018 | UAV123 | UAV123@10fps | Avg | Avg.FPS |
|---|---|---|---|---|---|---|---|---|---|
| **AVTrack-ViT** | | | (79.3, 62.4) | (77.0, 56.2) | (83.0, 62.7) | (83.2, 65.8) | (82.1, 64.8) | (80.9, 62.4) | 195.5 |
| | ✓ | | (**82.4**, **64.1**) | (**80.8**, **58.5**) | (**87.1**, **66.2**) | (**84.4**, **66.5**) | (82.9, 65.5) | (**83.5**, **64.2**) | - |
| | ✓ | ✓ | (81.3, 63.3) | (79.9, 57.7) | (86.4, 65.9) | (84.0, 66.2) | (**83.2**, **65.7**) | (82.9, 63.8) | **250.2** |
| **AVTrack-EVA** | | | (82.5, 63.0) | (79.1, 56.6) | (79.7, 60.7) | (80.4, 63.2) | (80.7, 63.3) | (78.9, 61.2) | 237.1 |
| | ✓ | | (**83.0**, **64.3**) | (**79.3**, **56.8**) | (**85.0**, **63.8**) | (**83.3**, **64.9**) | (**81.8**, **63.8**) | (**82.5**, **62.7**) | - |
| | ✓ | ✓ | (82.6, 64.0) | (78.8, 57.2) | (83.4, 62.5) | (83.0, 64.7) | (81.2, 63.5) | (81.8, 62.3) | **283.7** |
| **AVTrack-DeiT** | | | (84.2, 65.1) | (78.6, 56.7) | (81.6, 62.2) | (83.7, 66.1) | (82.7, 65.6) | (82.2, 63.2) | 208.4 |
| | ✓ | | (**84.8**, **65.5**) | (**83.2**, **59.4**) | (**86.4**, **65.6**) | (**85.3**, **67.1**) | (**83.7**, **66.3**) | (**84.7**, **64.8**) | - |
| | ✓ | ✓ | (84.3, 65.0) | (82.1, 58.7) | (86.0, 65.3) | (84.8, 66.8) | (83.2, 65.8) | (84.1, 64.3) | **256.8** |

*Table A3.* A evaluation of the effects of VIR and AM on the performance of the baseline trackers on DTB70, UAVDT, VisDrone2018, UAV123, and UAV123@10fps.

in Prec. and Succ. During this process, all baseline trackers undergo Avg. GPU speed enhancements of more than 20.0%, with AVTrack-ViT demonstrating an outstanding improvement of 26.0%.

## D. Application to SOTA Trackers

In Table A4, a comprehensive evaluation of the application to three SOTA trackers, i.e., ARTrack (Wei et al., 2023b), DropTrack (Wu et al., 2023), and GRM (Gao et al., 2023), is presented, encompassing all five datasets. Note that we substitute their original backbones with the tiny ViT, i.e., ViT-Tiny (Dosovitskiy et al., 2021), to save training time. In accordance with the statement made in the previous section, we only provide baseline speeds to eliminate any potential nuances caused by randomness. Specifically, when the VIR component is incorporated, the Avg. Prec. increases for ARTrack, DropTrack, and GRM are 2.6%, 1.7%, and 1.5%, respectively, while the Avg. Succ. increases are 1.7%, 1.1%, and 1.0%, respectively. After integrating AM, all baseline trackers witness GPU speed enhancements exceeding 14.0%, with DropTrack showcasing a remarkable improvement of 18.0%.

| Tracker | VIR | AM | DTB70 | UAVDT | VisDrone2018 | UAV123@10fps | Avg | Avg.FPS |
|---|---|---|---|---|---|---|---|---|
| **ARTrack(Wei et al., 2023b)** | | | (78.1, 59.8) | (**77.1**, **54.6**) | (77.7, 59.5) | (77.2, 60.8) | (77.5, 58.7) | 79.1 |
| | ✓ | | (**82.9**, **63.1**) | (76.2, 54.0) | (**79.9**, **60.8**) | (**81.2**, **63.8**) | (**80.1**, **60.4**) | - |
| | ✓ | ✓ | (82.1, 62.6) | (75.8, 53.7) | (79.4, 60.5) | (80.7, 63.5) | (79.5, 60.1) | **90.2** |
| **DropTrack(Wu et al., 2023)** | | | (80.7, 63.3) | (76.9, 55.9) | (81.5, 62.7) | (84.6, 66.6) | (80.9, 62.1) | 196.7 |
| | ✓ | | (**82.8**, **63.9**) | (**81.1**, **59.1**) | (**83.1**, **64.0**) | (**83.5**, **65.8**) | (**82.6**, **63.2**) | - |
| | ✓ | ✓ | (82.4, 63.6) | (80.5, 58.7) | (82.7, 63.7) | (82.3, 64.8) | (82.0, 62.7) | **232.3** |
| **GRM(Gao et al., 2023)** | | | (82.9, 64.3) | (79.0, 57.7) | (82.7, 63.4) | (83.2, 65.6) | (82.0, 62.8) | 213.3 |
| | ✓ | | (**84.8**, **65.2**) | (**80.1**, **58.6**) | (**84.1**, **64.6**) | (**84.8**, **66.8**) | (**83.5**, **63.8**) | - |
| | ✓ | ✓ | (84.4, 64.8) | (79.8, 58.3) | (83.7, 64.1) | (84.3, 66.5) | (83.1, 63.4) | **248.5** |

*Table A4.* A evaluation of the effects of AM and VIR on the performance of the baseline trackers on DTB70, UAVDT, VisDrone2018, UAV123, and UAV123@10fps

| $n_f$ | DTB70 | UAVDT | VisDrone2018 | UAV123 | UAV123@10fps | Avg. | Avg.FPS |
|---|---|---|---|---|---|---|---|
| 1 | (84.3, 65.0) | (**82.1**, 58.7) | (86.0, 65.3) | (84.8, 66.8) | (83.2, 65.8) | (84.1, 64.3) | **256.8** |
| 2 | (**84.8**, **65.3**) | (81.2, 58.1) | (86.3, 65.4) | (**85.4**, **67.1**) | (83.4, 65.9) | (84.2, 64,4) | **241.3** |
| 3 | (**85.0**, **65.4**) | (81.0, 58.0) | (86.5, 65.5) | (**85.0**, **66.9**) | (84.0, 66.2) | (84.3, 64.4) | **226.6** |
| 4 | (84.1, 64.9) | (**83.4**, **59.4**) | (87.0, 65.8) | (84.5, 66.6) | (83.6, 66.0) | (84.5, **64.5**) | 211.4 |
| 5 | (84.0, 64.8) | (**84.0**, **59.7**) | (87.1, 65.8) | (83.5, 65.9) | (**84.8**, **66.5**) | (**84.7**, **64.5**) | 193.5 |
| 6 | (84.5, **65.1**) | (81.9, 58.4) | (**87.4**, **66.0**) | (**85.1**, **67.0**) | (**84.3**, **66.3**) | (**84.6**, **64.6**) | 178.9 |
| 7 | (**85.1**, **65.4**) | (81.5, 58.2) | (**87.7**, **66.2**) | (84.2, 66.4) | (**84.1**, **66.2**) | (84.5, **64.5**) | 163.7 |
| 8 | (84.4, **65.1**) | (**84.0**, **59.8**) | (**88.0**, **66.3**) | (83.8, 66.1) | (84.0, 66.1) | (**84.8**, **64.7**) | 148.5 |

*Table A5.* Ablation study of the effect of layer $n_f$ (AM added only at layer $i > n_f$) on the Prec., Succ. and FPS of AVTrack on DTB70, UAVDT, VisDrone2018, UAV123, and UAV123@10fps.

## E. Study on When to Initiate the AM

In this section, we employ Table A5 to conduct a comprehensive analysis of the relationship between different initiating AM number of layers and performance. The evaluation is performed across all five datasets, with $n_f$ ranging from 1 to 8. As shown in the table, we observed that each incremental increase in $n_f$ results in a modest improvement in both Prec. and Succ., but also leads to a significant decrease of more than 5% in FPS. Taking into account the trade-off between precision and speed, we have opted to set the default value of $n_f$ as 1 in our implementation.

## F. Real-world Tests

We assess our AVTrack-DeiT and the baseline AVTrack-DeiT* on real-world tests by deploying them onto a standard UAV platform equipped with a Jetson AGX Xavier. Two example tracking results are shown in Fig. A1. As can be seen, our AVTrack-DeiT are able to track the two targets more accurately than the baseline in these examples where viewpoint change challenges are present. In order to test our method on a real drone, we integrated an embedded onboard processor, the NVIDIA Jetson AGX Xavier 32GB, into a typical UAV platform. In real-world UAV testing, the utilization rates of GPU and CPU are 39.7% and 13.5%, respectively, for AVTrack-DeiT, while those of the baseline are 43.3% and 16.7%, respectively. They remain at an average speed of 42.4 FPS and 36.7 FPS during the tests. Real-world testing on embedded systems directly verifies it can still maintain good robustness during viewpoint change and excellent performance and efficiency in various UAV specific challenges.

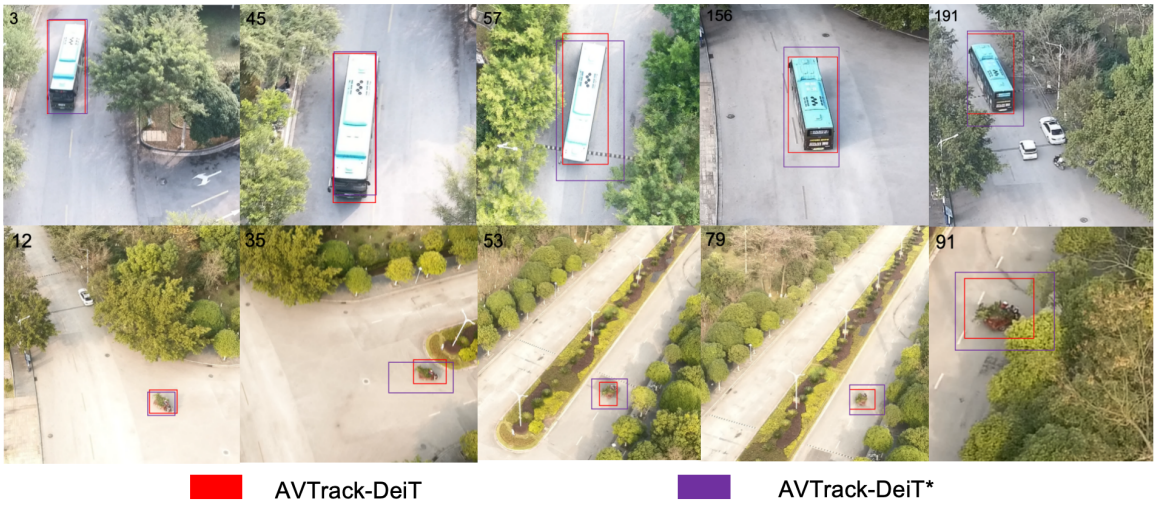


*Figure A1.* The actual data visualization captured from the UAV platform is presented, with the tracked target indicated by a bounding box. Various line representations illustrate target tracking across different environments, with the frame annotated in the upper left corner.

## G. Attribute-Based Evaluation

In order to demonstrate the superiority of our tracker against different trackers, i.e, DCF(Henriques et al., 2015), KCF (Henriques et al., 2015), SAMF (Li & Zhu, 2015), SAMF_CA (Mueller et al., 2017), STRCF (M. & et al., 2016), CN (Danelljan et al., 2014), HiFT (Cao et al., 2021), fDSST (Danelljan et al., 2017), MCCT_H (Wang et al., 2018), Staple (Bertinetto et al., 2016), Staple_CA (Mueller et al., 2017), SGDViT (Yao et al., 2023), ECO_HC (Danelljan et al., 2017), AutoTrack (Li et al., 2020b), ARCF (Huang et al., 2019), TCTrack (Cao et al., 2022), ABDNet(Zuo et al., 2023), UDAT (Ye et al., 2022b), RACF(Li et al., 2022a), P-SiamFC++ (Wang et al., 2022), and Aba-ViTrack (Li et al., 2023), we conducted an attribute-based comparison on three attribute subsets of UAVDT (row 1 to row 2), VisDrone 2018 (row 3 to row 4), and UAV123@10fps (row 5 to row 6), as depicted in Fig. A2. AVTrack-DeiT showcases remarkable performance in terms of Prec. and Succ. across the majority of these attributes. It is important to note that we also evaluate AVTrack-DeiT without implementing the components (AM and VIR), referred to as AVTrack-DeiT* for reference.

As shown in Fig. A2, taking the UAVDT dataset as an example (row 1 to row 2), our AVTracker exhibits optimal performance in terms of 'Large Occlusion', yet it secures the third and second positions in terms of 'Background Clutter' and 'Object Motion', respectively. Remarkably, across these three attributes, the integration of our proposed components leads to significant improvements when compared to AVTrack-DeiT*, achieving enhancements of 5.1%, 5.5%, and 8.5% in Prec., and 2.8%, 4.1%, and 5.8% in Succ., respectively. These results provide strong evidence for the effectiveness of our method in improving tracking performance.

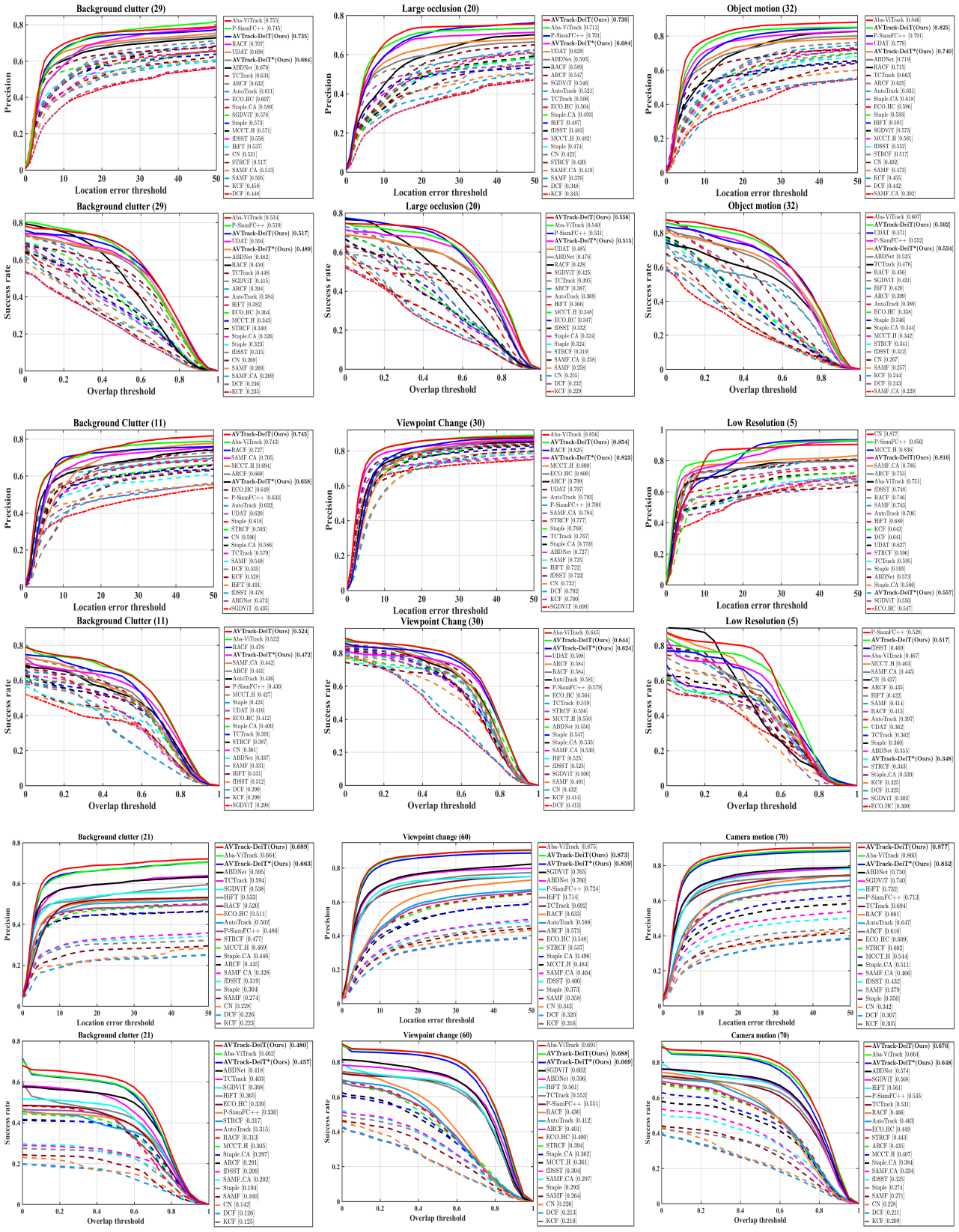


*Figure A2.* The precision plots and success plots of attribute-based comparison are presented for the attribute subsets of UAVDT (row 1 to row 2), VisDrone 2018 (row 3 to row 4), and UAV123@10fps (row 5 to row 6). Note that **AVTrack-DeiT*** denotes **AVTrack-DeiT** without the application of the proposed components.