Overlapped Trajectory-Enhanced Visual Tracking

Li Shen, Xuyi Fan, and Hongguang Li

***Abstract*— Deep-learning-based methods have achieved promising performance in visual tracking tasks. However, the backbones of the existing trackers normally emanate from the object detection realm, making them inefficient and insufficient in terms of spatial template matching. Moreover, such trackers apply temporal information without considering its authenticity during the online inference step, rendering them prone to error accumulation. To address these two issues, this work proposes OTETrack, a novel visual tracker with overlapped feature extraction and robust trajectory enhancement. The backbone of OTETrack, termed Overlapped ViT, slices the input image into overlapped patches to attain stronger template matching capabilities and sends them to alternating attention modules to maintain high model efficiency. Moreover, the trajectory enhancement mechanism in OTETrack is used to predict the center of the ladder-shaped Hanning window, which mildly penalizes the displacements between the spatial tracking results and the temporal predicted results to maintain the tracking consistency of a video sequence, thus mitigating the influences of spurious temporal information. Extensive experiments conducted on five benchmarks with thirteen baselines demonstrate the state-of-the-art performance of OTETrack. The source code and Appendix are released on https://github.com/OrigamiSL/OTETrack.**

***Index Terms*— Visual tracking, enhanced ViT-based tracking, trajectory-based tracking, Hanning window.**

I. Introduction

VISUAL tracking has been applied in practical systems covering a broad range of domains [1], [2], [3]; hence, it is one of the most important tasks in the computer vision (CV) realm. For handling sophisticated tracking conditions such as deformation, illumination variation, occlusion and background clutter in real-world practice, deep visual tracking methods [4], [5], [6], which are capable of leveraging neural networks with millions of parameters to extract the profound features of images, have become the prevailing approaches in recent years.

According to the theoretical basis of SiamFC [4], deep visual tracking models are commonly devised as matching models that successively locate a template in all images of an arbitrary video sequence. Notably, the image feature extraction ability of the utilized backbone is of paramount significance in visual tracking since it determines the quality of template matching results and significantly affects the ultimate tracking performance. The majority of visual tracking researchers [6], [7], [8], [9] are inclined to directly deploy the well-known backbones from the field of objection detection, e.g., ResNet [10] and ViT [11], in their networks with minimal modifications. Nevertheless, we notice that these approaches are not devised for visual tracking, and they need appropriate improvements to maximize their strengths in visual tracking, as do many works in other CV fields involving these prominent backbones [12], [13]. Therefore, it is feasible to employ the mature backbones developed in other CV fields for visual tracking; however, the characteristics of the visual tracking task must be considered to properly modify these methods and enhance their feature extraction capabilities and efficiency.

Moreover, the conventional deep visual tracking models [4], [14] are only able to make use of spatial information, which implies that temporal matching-based visual tracking is conducted for each frame. With the usage of better backbones [11], [15], such models may be better able to address partial sophisticated tracking conditions, e.g., deformation and illumination variations. However, they do not excel in cases with occlusion and background clutter, where the template is analogous to the surrounding environment or does not exist. Recently, some works [8], [9], [16], [17], [18], [19] have discovered that employing temporal information could solve the preceding problems. In such cases, these work leveraged historical trajectories [8], [9], [16] and features [17], [18], [19] to indicate the current locations of the target object. The contributions of these approaches will be briefly introduced in Section II-B. However, after investigating their methodologies and model designs, we discover that these methods ignore the fact that temporal information can be spurious in the inference phase. Thus, these approaches are prone to error accumulation in certain tracking occasions, e.g., those with camera vibrations and out-of-view targets, where the tracking failures are nearly ineluctable. Therefore, a more robust method for applying temporal information in visual tracking is needed.

To address the preceding issues, we propose a novel Overlapped Trajectory-Enhanced visual Tracking network (OTETrack). OTETrack possesses a revamped vision transformer (ViT), that is tailored for visual tracking as its backbone. This novel ViT, termed the Overlapped ViT, possesses additional patches, which are overlapped with the original isolated patches in the vanilla ViT. These additional patches act as connectors among the adjacent original patches,

Manuscript received 22 April 2024; revised 8 July 2024; accepted 2 August 2024. Date of publication 8 August 2024; date of current version 23 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62076019 and in part by the National Key Research and Development Program of China under Grant 2022YFB3904303. This article was recommended by Associate Editor J. Zhang. *(Li Shen and Xuyi Fan contributed equally to this work.) (Corresponding author: Hongguang Li.)*

Li Shen is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: shenli@buaa.edu.cn).

Xuyi Fan is with the School of Electronics and Information Engineering, Beihang University, Beijing 100191, China (e-mail: 18376480@buaa.edu.cn).

Hongguang Li is with the Beihang Institute of Unmanned System, Beihang University, Beijing 100191, China (e-mail: lihongguang@buaa.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCSVT.2024.3440330.

Digital Object Identifier 10.1109/TCSVT.2024.3440330

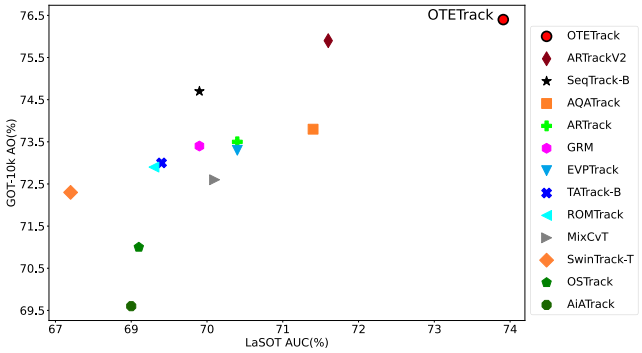


Fig. 1. The accuracy comparison on GOT-10k and LaSOT.

and the important features that cover multiple original patches can thus be extracted. To maintain the preceding computational magnitude, we propose alternating attention (ANA) to utilize the original and additional patches of the template and the search image in an alternating manner. The Overlapped ViT is composed of cascaded ANA modules to achieve improved efficiency. Since the targets in visual tracking tasks always locate around the search image center, the boundary features normally do not pertain to the target and can be omitted. Thus, the collective features of all additional patches effectively approximate the universal features and the rationality of ANA is guaranteed. Consequently, the feature extraction ability of Overlapped ViT is much stronger than that of the vanilla ViT but only with the expense of linear complexity. Furthermore, to exploit the temporal information in a more robust manner, OTETrack conducts trajectory prediction via exponential smoothing (ES) [20]. Though it is much simpler than deep forecasting methods, ES is more robust to nonstationarity trajectory prediction tasks due to its data-driven property. Its characteristic of placing more emphasis on later time series elements is also more suitable for visual tracking because the target motion is frequently irregular and nonstationary. The single ES (SES) method and Holt's ES (HES) method are applied according to the observed spatial tracking quality to make the trajectory prediction process of OTETrack more adaptive. The predicted target location is used as the center of the Hanning window, which is applied at the end of the model during the inference phase. This is done to penalize the large displacements between the spatial tracking results and the temporal prediction results. Hence, temporal information affects the spatial tracking results through the intermediate Hanning window, whose modality is indirect and more robust. Instead of mechanically using the conventional Hanning window, we transform it into a ladder-shaped window in which an area, rather than a single point, possesses the largest value, and the length of the peak area is dictated by the standard deviation of the trajectory. Therefore, the displacement penalty of the ladder-shaped Hanning (LH) window becomes more resistant to biased historical trajectory information. Leveraging the advanced mechanisms mentioned above, OTETrack achieves state-of-the-art performances, as shown in Fig. 1. Our main contributions can be summarized into the following four points:

- We propose a novel tracker with an ad-hoc backbone tailored for visual tracking and a new method that can robustly exploit temporal information.
- Leveraging the characteristics of visual tracking, we propose an Overlapped ViT, which is capable of efficiently extracting sufficient features from overlapped image patches via ANA.
- To cope with spurious temporal information during the inference phase, we incorporate the trajectory prediction task into an LH window, which mildly punishes the displacements between the spatial tracking results and the temporal prediction results.
- We conduct extensive experiments on five benchmarks to verify the promising tracking capability of OTETrack and the functions of its unique components in comparisons with thirteen cutting-edge baselines.

The remainder of this work is organized as follows. Section II introduces the recent developments related to deep visual tracking from the perspectives of spatial information extraction and temporal information application. Section III provides the preliminaries, including the definition of visual tracking, the basic architecture of one-stream visual tracking and two existing techniques for deep trackers. Section IV describes the architecture of OTETrack and its components in detail. The experiments conducted in Section V empirically demonstrate the state-of-the-art performance of OTETrack. Section VI summarizes this work and discusses further research directions.

II. Related Works

A. Deep Visual Trackers

The majority of the available deep visual trackers are template matching-based networks, and their basic architecture was devised by [4]. Their tracking pipelines are briefly described with three components: (1) a backbone for extracting the spatial features of the search image and the associated template; (2) a neck for matching the feature maps of the search image and the template, which is not always necessary since many trackers [6], [19], [21] simultaneously perform the feature extraction and matching processes; and (3) a head for obtaining the spatial tracking result, i.e., the target location, via the matching result. The observed temporal information can also be involved in these steps; however, this aspect is discussed in Section II-B. Notably, the quality of spatial tracking results is predominantly determined by the image feature extraction capabilities. However, we notice that nearly all trackers are inclined to employ the existing backbones from other CV fields, e.g., ResNet in [1] and [7] and ViT in [5], [6], [8], and [9]. Unfortunately, it is not efficient to directly apply the advanced backbones from other CV fields since they are not tailored for visual tracking. The best evidence of this is that SwinTransformer [15] achieves better performance than the vanilla ViT in the realm of object detection, whereas plenty of tracking models [8], [9] with the ViT as their backbones outperform SwinTrack [16], which adopts SwinTransformer as its backbone. MixFormerV2 [22], which is built upon ViT, also outperforms its previous version MixFormer [21], which

was based on more advanced CvT (MixCvT) [23]. Therefore, studying a way to improve the mature backbones derived from other CV realms to fit the visual tracking task, instead of blindly pursuing the state-of-the-art methods, is the key to maximizing their visual tracking power.

Different from those who simply utilize the existing feature extraction backbones without modifications in their models, some researches have attempted to exploit them in a more efficient manner. For instance, Ye et al. [6] proposed a type of one-stream visual tracking framework based on ViT. MixFormerV2 [22] even completely mixes image features and the target location in ViT to further simplify the tracking pipeline. Also based on ViT, GRM [14] further utilizes an adaptive method to categorize the patch tokens in the search image to separate the unnecessary background tokens in advance. Similar idea is adopted by PATrack [3] but its categorization is built upon its proposed probabilistic assignment approach. The uses of the existing backbones by these methods are miscellaneous, but the formulas of these backbones are unchanged. Their template matching processes may be simplified and more efficient, but the feature extraction capabilities of the employed backbones are still not improved. In contrast, our proposed Overlapped ViT leverages the characteristic of visual tracking to rationally enhance its feature extraction capability through the efficient extraction of more abundant features from overlapped patches with ANA modules.

B. Visual Tracking With Temporal Information

Apart from reinforcing the ability to extract features from search images and templates, many researchers have developed diverse methods to enhance visual tracking from the viewpoint of temporal information. These methods can be divided into two groups based on how they use temporal information: (1) feature-based methods and (2) trajectory-based methods.

Feature-based methods incorporate historical image features into their models since they can provide historical target features and their relations with the background. This helps the models accurately locate the target in the current search image and refrain from encountering tracking failures caused by background clutter. To achieve this goal, TATrack [19] simultaneously sends the previous search image, the current search image and the template to its tracking network. EVPTrack [17], AQATrack [18] and MT-Track [24] draw upon more previous search image features via recursive causal transmission. Cai et al. [25] paid attention to the potential contextual appearance changes between adjacent frames so that the previous variation tokens could be further considered in the next frame by their proposed ROMTrack.

Trajectory-based methods [2] attempt to leverage historical trajectory information to guide their models to continue tracking the target. Although trajectory information may not be as abundant as image features, applying it in trackers is much less expensive. To apply trajectory information, SwinTrack [16] concatenates the embedded trajectory information with the feature map acquired by template matching. Imitating Pix2Seq [26], ARTrack [9] applies the tokenization technique to avoid parameter explosion and translates the entire target trajectory into a single sequence. Then, the target location coordinates can be causally and recursively deduced.

In summary, no matter whether these models are feature-based or trajectory-based, they fully apply temporal information in their tracking networks without considering its authenticity during the inference phase. Consequently, these models suffer from error accumulation problems caused by tracking errors or tracking failures, once spurious temporal information is encountered.

In this work, the temporal information is applied via trajectory prediction to avoid overloading the model. Moreover, the trajectory prediction results affect the spatial tracking results through an intermediate Hanning window; this is a milder strategy than directly mixing the results, thus mitigating the error accumulation problem. The proposed trajectory prediction method is exquisitely designed, and the Hanning window is revamped to further enhance its robustness.

III. Preliminary

A. Problem Statement

Given a template $\mathbf{Z}$ and a video sequence $\boldsymbol{\mathcal{X}} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_L\}$, the visual tracking task involves tracking the locations of an object (objects) $\mathbf{Z}$ in $\boldsymbol{\mathcal{X}}$. L is the number of frames in the video. Each frame $\mathbf{X}_i$ is termed a search image. This work focuses on single-object tracking; therefore, $\mathbf{Z}$ merely has one target object, as does $\mathbf{X}_i$. Following the commonly agreed-upon method [4], the template is cropped based on the location of the target center itself, and the search image is cropped based on the location of the target center in the previous frame. Consequently, the target definitely lies at the center of the template and is normally near the center of the search image because the target motion between adjacent frames is normally minor. This work also takes historical trajectory information into account; thus, the proposed model takes the following form:

$$[x^t_{min}, x^t_{max}, y^t_{min}, y^t_{max}] = \mathrm{f}(\mathbf{T}_{t-h:t-1}, \mathbf{X}^t, \mathbf{Z}_1, \mathbf{Z}^t_2; \theta) \quad (1)$$

where $\mathbf{T}_t = [x^t_{min}, x^t_{max}, y^t_{min}, y^t_{max}]$ are the four target corner coordinates of the current search image at time t, $\mathrm{f}(\cdot)$ denotes the model and θ represents its learnable parameters. $\mathbf{T}_{t-h:t-1}$ denotes the trajectory coordinates in the previous h frames, $\mathbf{X}^t$ is the current search image, $\mathbf{Z}_1$ is the static template and $\mathbf{Z}^t_2$ is the dynamic template at time t, which is elucidated in Section III-D.

B. One-Stream Visual Tracking

With the ability to simultaneously extract image features and conduct template matching in the backbone, one-stream visual tracking [6] has become more prevalent than two-stream visual tracking [4], which performs template matching after separately extracting the features of the search image and the template, in recent years. Leveraging the convenience of attention mechanisms and ViT, the backbones of one-stream visual trackers receive a hybrid form of the patches derived from the search image and the template; thus, the feature extraction process involving these two images and the template matching can be performed concurrently. Due to its success, one-stream tracking is also utilized in OTETrack.

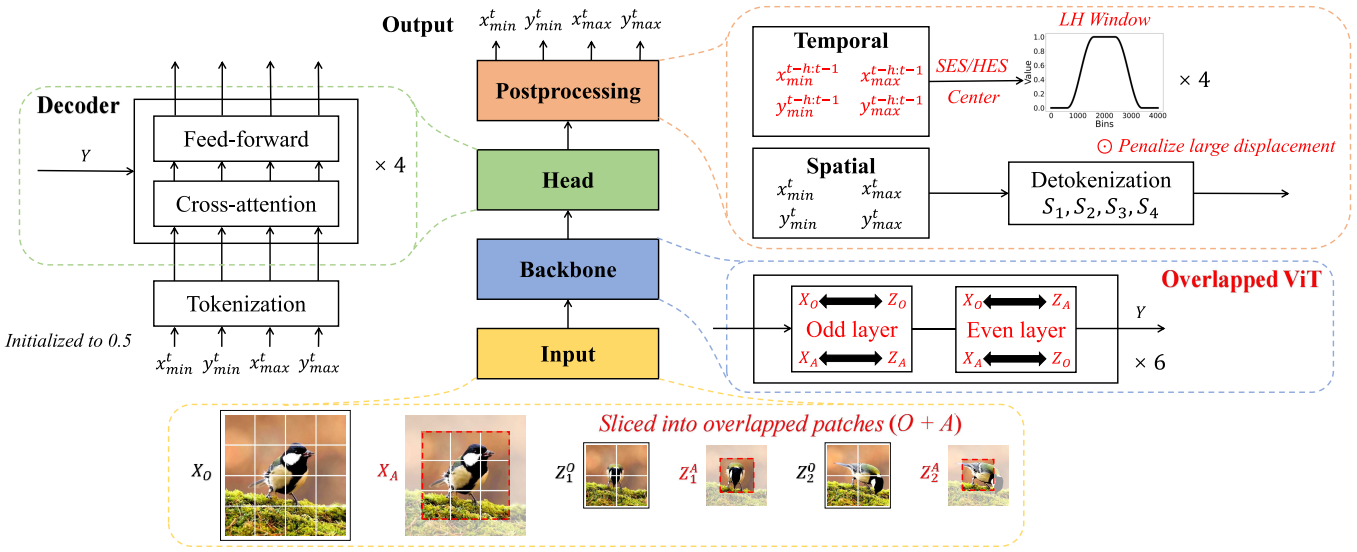


Fig. 2. The overview architecture of OTETrack. The input search image, static template and dynamic template are all sliced into overlapped patches, whose formulas are shown in Fig. 3. The concrete architectures of six odd/even layers in Overlapped ViT are shown in Fig. 4. The previous trajectory with the length h is employed to predict the centers of four LH windows, whose usage will be elaborated in Section IV-D. All novel components are highlighted in red.

C. Tokenization

Imitating ARTrack [9], [27], we tokenize [26] the four corner coordinates of the target object in the search image during the embedding process; therefore, they are all transformed to integers belonging to $[1, n_{bins}]$. n_{bins} is the number of bins, which needs to be larger than the image height and width to achieve zero quantization error. A shared and learnable vocabulary $\mathbf{v}$ is used to represent each bin value via the corresponding word vector. Thus the tokenized target location coordinates can always be found in this specific vocabulary. The ultimate network tracking result is composed of the word vectors derived from the four corner coordinates, which are transformed into the real target location coordinates via detokenization.

D. Template Updating

For better handling the deformation problem, online template updating [28] is a feasible and convenient strategy. Similar to SeqTrack [8], we employ two templates in OTETrack. Although both are initialized with the target object in the first frame, one of them is static ($\mathbf{Z}_1$), and the other is dynamic ($\mathbf{Z}_2^t$) during the online inference process. The dynamic template updates itself with the qualified spatial tracking results at fixed intervals. The significance of the matching score during the detokenization process is used to evaluate whether a certain tracking result is qualified. More details are given in Sections IV-C and V-B3.

E. Hanning Window

The Hanning window is commonly used in visual tracking methods to penalize large displacements between two successive frames, and its formula is as follows:

$$\mathrm{H}(n) = \begin{cases} 0.5 \times [1 - \cos(\frac{2\pi n}{N-1})], & \text{if } 0 \le n \le N-1 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where N represents the size of the Hanning window $\mathrm{H}(n)$. For this purpose, the previously developed methods [5], [8], [29] set the coordinates of the previous frame as the center of the Hanning window to penalize distant output coordinates with significant confidence. However, this approach is only appropriate for high-quality video with sufficient frames per second (FPS) or tracking circumstances with slow motions, which are not always encountered in real-world practice.

IV. Methodology

Before introducing of the novel components contained in OTETrack, we sketch its architecture in Fig. 2. The associated pipeline can be summarized in three steps. (1) The search image, static template and dynamic template are sliced into overlapped patches and sent to the Overlapped ViT, which simultaneously performs feature extraction and template matching. (2) A decoder is used as the head to deduce the word vectors of four tokenized target corner coordinates in parallel via the features provided by the Overlapped ViT and match them with the vocabulary. (3) The historical trajectory prediction result is used to form four LH windows for reorganizing the matching scores and obtaining the final tracking result. More details concerning these three steps and the various components of OTETrack are described below.

A. Overlapped ViT

The vanilla ViT slices images into nonoverlapped patches, and these patches are merely built connections via attention. This means that the features of each patch are treated as a

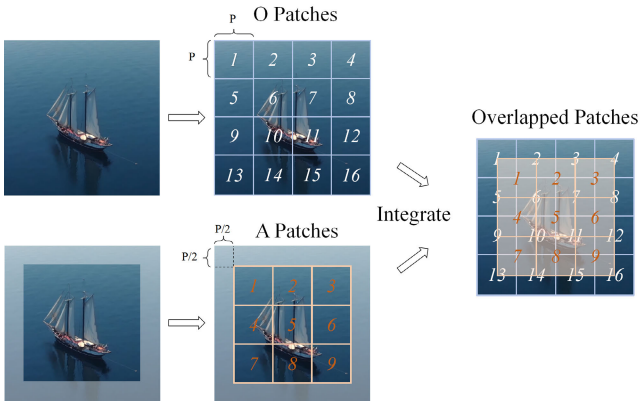


Fig. 3. The formula of slicing a image into O patches and A patches. The first row uses the whole image to slice into non-overlapped O patches. The second row cuts down the boundary parts, whose lengths are half the patch size P, of the image and use the rest to slice into non-overlapped A patches. However, the A patches are overlapped with the O patches, which compensates the cross-patch features of O patches.

whole and are collectively performed attentions with other patches in the ViT. Consequently, cross-patch features are difficult for the ViT to extract. To strengthen the feature extraction ability of the ViT, we design an Overlapped ViT, which slices the input image into overlapped patches, as shown in Fig. 3. We term the patches, which are also possessed by the vanilla ViT, O patches. The additional patches, which are overlapped with the O patches, are denoted as A patches for brevity in the remainer of this work. The A patches bridge the original isolated O patches of the ViT; thus, the cross-patch features become easier to extract. Note that the A patches do not overlap with each other and that their collective features merely dismiss the boundary areas of the image. However, these areas have the greatest probabilities of being the background since the target often locates near the image center in a visual tracking task. This means that the collective features of the A patches highly approximate the universal features of the entire image. This property is quite significant for the theoretical foundation of the ANA modules in the Overlapped ViT.

B. Alternating Attention

Making use of more patches is a common way [30], [31], [32], [33] to enhance the feature extraction capabilities of ViT in CV. The core difficulty is how to efficiently wield these patches in terms of the characteristics of the downstream tasks. If not cautiously handled, e.g., blindly performing attention, which owns quadratic complexity, to all patches, the extra computation expense would be unacceptable. Therefore, we propose ANA to address this challenge. As shown in Fig. 4, each ANA mechanism is composed of two standard multihead self-attention (MHSA) modules to which the patches are fully allocated. The formula of the MHSA process is as follows:

$$\begin{aligned}\text{Head}_i(\mathbf{Q},\mathbf{K},\mathbf{V}) &= \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{C_i}})\mathbf{V}, i=1,2,\ldots,N \\ \text{MHSA}(\mathbf{Y}) &= \text{Concat}([\text{Head}_i(\mathbf{Y}\mathbf{W}_i^Q,\mathbf{Y}\mathbf{W}_i^K,\mathbf{Y}\mathbf{W}_i^V)])\mathbf{W}\end{aligned} \tag{3}$$

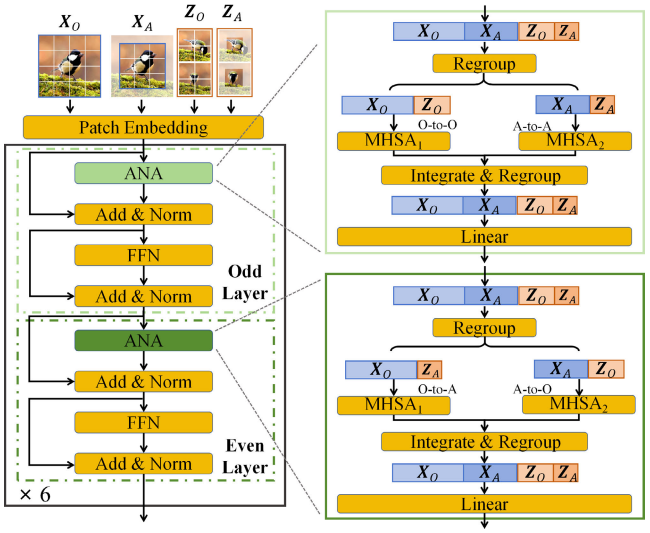


Fig. 4. The architecture of Overlapped ViT with six odd layers and six even layers. Each layer is composed of an ANA module and a feed-forward layer. The patch allocation modalities to the ANA modules of odd and even layers are different, as shown in the right part of the figure.

where $\mathbf{Y} \in \mathbb{R}^{P\times C}$ is the input of the MHSA mechanism, P denotes the number of patches and C denotes the channel size. N is the number of heads. $\mathbf{Q},\mathbf{K},\mathbf{V} \in \mathbb{R}^{P\times C_i}$ are the query vector, key vector and value vector, respectively, which are obtained via separate linear projections $\mathbf{W}_i^Q,\mathbf{W}_i^K,\mathbf{W}_i^V \in \mathbb{R}^{C\times C_i}$ with $\mathbf{Y}$. Following the vanilla ViT, the number of heads is 12, and the head channels are uniformly allocated; i.e., $C_i \equiv C//12$. $\mathbf{W} \in \mathbb{R}^{C\times C}$ is the output linear projection matrix of ANA.

However, the patch allocation modalities in the odd and even layers are different. The first MHSA mechanism in any odd layer receives the O patches of the search image ($\mathbf{X}_O$) and their templates ($\mathbf{Z}_O$), while the A patches of the search image ($\mathbf{X}_A$) and their templates ($\mathbf{Z}_A$) are sent to the second MHSA mechanism. In the even layers, the positions of the O and A patches of the templates are interchanged relative to those in the odd layers, which means that the first MHSA mechanism now receives the $\mathbf{X}_O$ and $\mathbf{Z}_A$, while the second MHSA mechanism now receives the $\mathbf{X}_A$ and $\mathbf{Z}_O$. Each ANA module is followed by a feedforward layer akin to the canonical ViT layer, as follows:

$$\text{FFN}(\mathbf{Y}) = \text{GELU}(\mathbf{Y}\mathbf{W}_1)\mathbf{W}_2 \tag{4}$$

where $\text{FFN}(\cdot)$ denotes the feedforward layer, $\text{GELU}(\cdot)$ is a prevailing activation function [34], $\mathbf{W}_1 \in \mathbb{R}^{C\times 4C}$ and $\mathbf{W}_2 \in \mathbb{R}^{4C\times C}$. Thus, the entire process is as follows:

$$\begin{aligned}\text{ANA}(\mathbf{Y}) &= \text{Concat}(\text{MHSA}_1(\mathbf{Y}_1),\text{MHSA}_2(\mathbf{Y}_2)) \\ \mathbf{Y}_0 &= \text{Norm}(\mathbf{Y}+\text{ANA}(\mathbf{Y})) \\ \mathbf{Y}_{out} &= \text{Norm}(\mathbf{Y}_0+\text{FFN}(\mathbf{Y}_0))\end{aligned} \tag{5}$$

where $\mathbf{Y}_1$ and $\mathbf{Y}_2$ denote the patches allocated to the first mechanism MHSA_1 and the second mechanism MHSA_2, respectively. $\text{Norm}(\cdot)$ is the layer normalization operation. $\mathbf{Y}_{out}$ is the final output of this layer.

The alternating interactions between the O patches and A patches in the search image and the templates strengthen

the templating matching effect relative to that of the vanilla ViT in that the matching processes additionally involve the A patches, which represent the cross-patch features of the O patches. The O patches and the A patches obtained from identical sources (e.g., $\mathbf{X}_O$ and $\mathbf{X}_A$ both originate from the search image $\mathbf{X}$) are not allowed to interact with each other throughout the network, which makes the process of extracting the inner features of the search image or templates not as strong as directly performing the attention to all patches. However, this design is a consequence of good efficiency, as we want to reduce the computational complexity to the greatest extent while guaranteeing that the cross-patch features represented by the A patches are fully considered in the most significant visual tracking step (i.e., the template matching process) instead of performing pure image feature extraction. Moreover, as the collective A patches have quasiuniversal features, any MHSA mechanism in any ANA module receives intact information from the search image and the templates. Furthermore, the number of patches contained in each MHSA mechanism is identical to or smaller than that of any ViT-based one-stream tracking framework, e.g., OSTrack. Thus, the additional complexity is solely caused by the doubled number of MHSA mechanisms, which is obviously a linear increase. Suppose that the numbers of O patches and A patches in the search image are N_O and N_A, respectively, while the numbers of O patches and A patches in each template are M_O and M_A, respectively. Then, the computational complexities of the MHSA_{ViT} mechanism in the original ViT, which receives only the O patches, and the ANA module are as follows:

$$\begin{aligned}
\text{Complexity}(\text{MHSA}_{ViT}) &= 2 \times (N_O + 2 \times M_O)^2 \times C \\
\text{Complexity}(\text{ANA}_{odd}) &= 2 \times (N_O + 2 \times M_O)^2 \times C \\
&\quad + 2 \times (N_A + 2 \times M_A)^2 \times C \\
&< 4 \times (N_O + 2 \times M_O)^2 \times C \\
&< 2 \times \text{Complexity}(\text{MHSA}_{ViT}) \\
\text{Complexity}(\text{ANA}_{even}) &= 2 \times (N_O + 2 \times M_A)^2 \times C \\
&\quad + 2 \times (N_A + 2 \times M_O)^2 \times C \\
&< 4 \times (N_O + 2 \times M_O)^2 \times C \\
&< 2 \times \text{Complexity}(\text{MHSA}_{ViT})
\end{aligned} \tag{6}$$

Then, the linearity of the additional complexity is proven. Consequently, the participation of ANA in the Overlapped ViT ensures a good tradeoff between complexity and efficiency.

C. One-Step Decoder

OTETrack employs a simple decoder as its head to generate the spatial tracking result. This decoder receives four tokenized target corner coordinates $[x^t_{min}, y^t_{min}, x^t_{max}, y^t_{max}]$ and the output of the Overlapped ViT $\mathbf{Y} \in \mathbb{R}^{P \times C}$. $\mathbf{Y}$ contains the ultimate feature maps of the search image and two templates.

To obtain the tokenized corner coordinates, $T_t = [x^t_{min}, y^t_{min}, x^t_{max}, y^t_{max}]$, which are initialized to 0.5 since they are certainly unknown at the beginning, are first transformed into $n_{bins}//2$ via a uniform discretization function $U_d : [0, 1] \rightarrow [1, n_{bins}]$. Then, the word vector $\mathbf{v} \in \mathbb{R}^D$ representing $n_{bins}//2$ can be found as the vocabulary $\mathbf{V} \in \mathbb{R}^{n_{bins} \times D}$, which possesses the word vectors describing all integer numbers between $[1, n_{bins}]$, for representing $[x^t_{min}, y^t_{min}, x^t_{max}, y^t_{max}]$ as $\mathbf{w} = [\mathbf{v}, \mathbf{v}, \mathbf{v}, \mathbf{v}] \in \mathbb{R}^{4 \times D}$. 0.5 is selected as the initialization value because the target is most likely to be located at the image center, and the target location range is normalized to $[0, 1] \times [0, 1]$. A linear projection layer $\mathbf{W}_{tr} \in \mathbb{R}^{C \times D}$ is employed for $\mathbf{Y}$ to change its hidden dimensionality from C to D; therefore, cross-attention can occur in the decoder.

As shown in Fig. 2, the decoder in OTETrack is composed of four layers, each containing a cross-attention module and a feedforward layer, to deduce the spatial tracking results via $\mathbf{Y}$. The cross-attention process is analogous to Eq. (3). The only difference is that the query vector is obtained by $\mathbf{w}$, while the key vector and the value vector are obtained by $\mathbf{Y}$. The concrete process is omitted for brevity, and the output of the last decoder layer is denoted by $\mathbf{w}_{out} \in \mathbb{R}^{4 \times D}$. Unlike SeqTrack [8] and ARTrack [9], which recursively deduce the target coordinates, OTETrack abandons the causal relationships of the target coordinates to avoid the time-consuming multistep autoregression process. Therefore, the spatial tracking result of OTETrack is acquired in one step via detokenization which matches $\mathbf{w}_{out}$ with the word vectors in $\mathbf{V}$ and takes those with the largest matching scores, as follows:

$$\begin{aligned}
&\textbf{Score} \\
&= \text{softmax}(\mathbf{w}_{out}\mathbf{V}^\top) = [\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_4] \\
&[x^t_{min}, y^t_{min}, x^t_{max}, y^t_{max}] \\
&= [\text{argmax}(\mathbf{S}_1), \text{argmax}(\mathbf{S}_2), \text{argmax}(\mathbf{S}_3), \text{argmax}(\mathbf{S}_4)]
\end{aligned} \tag{7}$$

where the softmax function in Eq. (7) normalizes the matching scores to [0, 1], $\mathbf{S}_i \in \mathbb{R}^{n_{bins}}$ denotes the matching score distribution and the argmax$(\cdot)$ function seeks the position with the largest matching score. Then, the target location in the image is obtained via a uniform continuation: $U_c : [1, n_{bins}] \rightarrow [0, 1]$. It has been empirically demonstrated [8] that a spatial tracking result is reliable if its largest matching scores are significant, which means that they are much larger than the remaining scores in $\mathbf{S}_i$. If this is achieved, the spatial tracking result can be used to crop the current search image and then update the dynamic template with the current cropped search image. This procedure is described as follows:

$$\begin{aligned}
S &= \Sigma(\text{argmax}(\mathbf{S}_i)) \\
\mathbf{Z}^t_2 &= \begin{cases} \text{Cropped}(\mathbf{X}), & \text{if } S > \eta \\ \mathbf{Z}^{t-1}_2, & \text{otherwise.} \end{cases}
\end{aligned} \tag{8}$$

where η is the matching score threshold.

D. Robust Usage of Temporal Information

To robustly exploit the temporal information, OTETrack integrates the Hanning window into the trajectory prediction procedure. Recall that the conventional Hanning window penalizes a large displacement between the tracking result $\mathbf{T}_t$ and the target location of the last frame $\mathbf{T}_{t-1}$ by recalculating the confidence scores (matching scores in this work) of the location candidates with the Hanning window. This method fails to capture longer historical information; however, it is

milder and more robust since it merely constrains the spatial tracking result within a certain area around $\mathbf{T}_{t-1}$. Conversely, the methods described in Section II-B directly mix the temporal information with the spatial tracking results, which leads to error accumulation and even the eternal loss of the target once a tracking failure occurs. To exert the advantages of these two methods and overcome their weaknesses, OTETrack employs four Hanning windows $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4]$ to separately postprocess the matching score distributions $[\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_4]$ of the four corner coordinates $\mathbf{T}_t$; however, the window centers are predicted by the trajectories $\mathbf{T}_{t-h:t-1}$. Therefore, $\mathbf{H}$ can penalize the displacements between the spatial tracking result and the trajectory prediction result. The new matching score distributions $\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4]$ are obtained by applying the Hadamard product $\odot$ to $\mathbf{H}_i$ and $\mathbf{S}_i$ as follows:

$$\mathbf{M}_i = \mathbf{H}_i \odot \mathbf{S}_i, \quad i = 1, 2, 3, 4 \tag{9}$$

Note that the dynamic template is still updated according to the old S, which represents the quality of the spatial tracking process.

We exquisitely design the prediction method and modify the format of Hanning window to further alleviate the error accumulation problem encountered during trajectory prediction. We employ ES [20] as the forecasting method due to its simplicity and convenience. The main property of ES is that it predicts the future with weighted input elements for which later elements are given more weights. This greatly fits real-world visual tracking scenarios, where the target motion is nonstationary and irregular. The η in Eq. (8) is also used here to determine the type of applied ES method. When the spatial tracking result is reliable (i.e., $S > \eta$), HES, which seeks a linear function to fit the trajectory trend and makes the prediction of target, is adopted to penalize the displacements of the spatial tracking results and the historical trajectory trend. The background clutter problem suffered by the conventional Hanning window is thus mitigated. Taking the HES prediction of x_{min}^t as an example, the process is as follows:

$$\begin{aligned} \text{Forecasting} : \hat{x}_{min}^{t|t-h} &= l_t + kd_t \\ \text{Level} : l_t &= b_1 x_{min}^{t-1} + (1-b_1)(l_{t-1} + d_{t-1}) \\ \text{Trend} : d_t &= b_2(l_t - l_{t-1}) + (1-b_2)d_{t-1} \end{aligned} \tag{10}$$

where $\hat{x}_{min}^{t|t-h}$ is the h-step-ahead forecast obtained after h iterations with $x_{min}^{t-h:t-1}$. b_1 and b_2 are the smoothing parameters used to control the level equation and the trend equation, respectively. Specifically, the forecasting equation states that forecasting is conducted by incrementing the last estimated level l_t by k times the last trend d_t. l_t is obtained via a weighted average of the previous observation x_{min}^{t-1} and the summation of the previous level and trend $(l_{t-1} + d_{t-1})$. The trend equation is used to deduce d_t via a weighted average of the successive difference $(l_t - l_{t-1})$ and the previous trend (d_{t-1}).

In contrast, the conservative SES method, which simply predicts the target location via the weighted averages of the trajectory, is used when $S < \eta$ to constrain the target location within the historical trajectory distribution. Taking x_{min}^t as an example, the process is as follows:

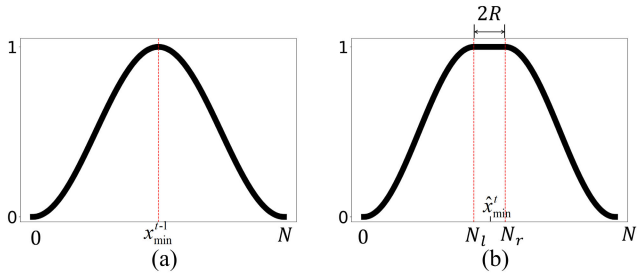


Fig. 5. The comparison of the Hanning window formulas on the application to x_{min}^t. (a) A conventional Hanning window H(n) with the previous x_{min}^{t-1} as the window center. (b) A Ladder-shaped Hanning window LH(n) with the predicted $\hat{x}_{min}^t$ as the window center.

$$\begin{aligned} \text{Forecasting} : \hat{x}_{min}^{t|t-h} &= l_t \\ \text{Level} : l_t &= a x_{min}^{t-1} + (1-a) l_{t-1} \end{aligned} \tag{11}$$

which is equivalent to Eq. (10) without the trend equation and the trend term. As the template matching quality is supposed to be low in this scenario, which can be caused by various factors (e.g., out-of-view and low resolution problems), the Hanning window, whose center is predicted by the conservative SES method, is capable of preventing the tracking result from jumping from one location to another. Hence, the error accumulation problem, to which trajectory-based visual tracking is prone, is alleviated when a tracking failure occurs or when the trajectory is imprecise.

Regarding for the modality of the Hanning window, its conventional center, which has a peak value of 1 in Eq. (2), is extended to a peak area with an additional $R = \beta \times std$ at each side to form the LH window, as follows (with x_{min}^t as an example):

$$\text{LH}(n) = \begin{cases} \text{H}(n), & \text{if } n < N_l \\ 1, & \text{if } N_l \le n \le N_r \\ \text{H}(n-2R), & n > N_r \end{cases} \tag{12}$$

where $\hat{x}_{min}^t$ is predicted based on $x_{min}^{t-h:t-1}$ via HES or SES, β is the factor that controls the peak area and std is the standard deviation of $x_{min}^{t-h:t-1}$. N is identical to N in Eq. (2), $N_l = \frac{N}{2}$ and $N_r = \frac{N}{2} + 2R$. Because the trajectory can be erroneous, the LH window makes the ground truth of the target location more likely to be contained within the peak area; thus, the spatial tracking result is less likely to be misled by the temporal tracking result if pathological prediction occurs. A comparison between a conventional Hanning window and the window used in OTETrack is shown in Fig. 5. Consequently, the robustness of the temporal information usage method in OTETrack is further enhanced.

E. Loss Function

Following ARTrack [9], OTETrack is trained to maximize the log-likelihood of the tokenized corner coordinates and the tokenized ground truths via a softmax cross-entropy function, and the similarity between the predicted bounding box and the corresponding ground truth via SIoU [35]. Hence, the loss

TABLE I
THE NUMERICAL DETAILS OF ALL BENCHMARKS

Benchmarks	Sequence Numbers (Train Split)	Sequence Numbers (Test Split)	Classes
GOT-10k	10000	180	563
TrackingNet	30132	511	21
LaSOT	1120	280	70
LaSOT$_{ext}$	-	150	15
UAV123	-	123	9

'-' indicates that certain benchmark does not own a train split.

TABLE II
THE ATTRIBUTES OF ALL BASELINES

Baselines	Sources	Backbones
OTETrack$_{256}$	Ours	Overlapped ViT
ARTrackV2$_{256}$ [27]	CVPR'24	ViT
AQATrack$_{256}$ [18]	CVPR'24	HiViT
EVPTrack$_{224}$ [17]	AAAI'24	HiViT
ARTrack$_{256}$ [9]	CVPR'23	ViT
SeqTrack-B$_{256}$ [8]	CVPR'23	ViT
GRM$_{256}$ [14]	CVPR'23	ViT
ROMTrack$_{256}$ [25]	ICCV'23	ViT
TATrack-B$_{256}$ [19]	AAAI'23	LCA+SwinTransformer
MixFormerV2-B$_{288}$ [22]	NIPS'23	Distilled ViT
MixCvT$_{320}$ [21]	CVPR'22	CvT
OSTrack$_{256}$ [6]	ECCV'22	ViT
AiATrack$_{320}$ [7]	ECCV'22	ResNet
SwinTrack-T$_{224}$ [16]	NIPS'22	SwinTransformer

function used in the training phase is shown in Eq. (13), where λ_1 and λ_2 are the weights of the two losses. We set $\lambda_1 = 2$ and $\lambda_2 = 2$ in our experiments.

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{ce} + \lambda_2 \mathcal{L}_{SIoU} \tag{13}$$

V. EXPERIMENTS

We perform extensive experiments to evaluate the comprehensive performance of OTETrack. Section V-A introduces the datasets and the baselines used in the experiments. Section V-B provides the implementation details. The general tracking performances achieved by OTETrack and the other thirteen baselines on four prevailing benchmarks are shown and compared in Section V-C. Section V-D compares both the tracking accuracy and the tracking efficiency levels of OTETrack and the other trackers. The novel components of OTETrack are individually evaluated in the ablation study conducted in Section V-E. In Section V-F, a parameter sensitivity analysis is performed to demonstrate that spatial tracking quality is a suitable discriminator for ES methods and that OTETrack is capable of handling long historical trajectories. The case studies presented in Section V-G visualize the performance of OTETrack. There are also plenty of additional results and analysis involving more comprehensive experiments on all benchmarks and more case studies in Appendix of the Supplementary Material.

A. Datasets and Baselines

1) Datasets: To unveil the tracking ability of OTETrack, we adopt five prevailing benchmarks: TrackingNet [36], LaSOT [37], LaSOT$_{\text{ext}}$ [38], GOT-10k [39] and UAV123 [40]. The numerical details are given in Table I.

2) Baselines: Thirteen state-of-the-art baselines that were proposed in the past two years are used in our experiments to highlight the superior performance of OTETrack. Their significant characteristics, as well as those of OTETrack, are shown in Table II. Most of these baselines employ existing mature backbones derived from the object detection field, including ResNet [10], ViT [11], CvT [23], SwinTransformer [15] and HiViT [41], as their backbones. The only two modified approaches, the LCA modules in TATrack and the distilled ViT in MixFormerV2, are used for materializing the correlations among multiple images and achieving better efficiency, respectively. Therefore, the characteristics of visual tracking are still not used in these backbones to efficiently improve the template matching capabilities of these methods, as in the overlapped ViT. In effect, not all of these baselines have provided results obtained on the employed datasets. We supplement the missing results via their released source codes and the default weight files, unless their authors do not provide them (MixFormerV2 does not provide the weight for GOT-10k. The source code of ARTrackV2 was not released prior to the submission of this work to the journal.).

B. Implementation Details

1) Model Configuration: To conduct a fair comparison with the above ViT-based trackers, the Overlapped ViT shares identical learnable parameters with ViT-Base. This means that the only difference between the Overlapped ViT and ViT-Base is the attention module, where the Overlapped ViT employs ANA and ViT-Base employs canonical attention. The variant of the OTETrack model used in the experiment is termed OTETrack$_{256}$. The image sizes for the templates and search images are 128×128 and 256×256, respectively. The patch size is 16×16, which is identical to that of ViT-Base. This means that the numbers of O patches and A patches in the search image are $16 \times 16 = 256$ and $15 \times 15 = 225$, respectively, while the numbers of O patches and A patches in the templates are $8 \times 8 = 64$ and $7 \times 7 = 49$, respectively. For those baselines that do not have variants with identical configurations, we choose variants whose image sizes are similar, e.g., MixFormerV2-B$_{288}$, to ensure that any model tested in the experiment does not apparently benefit from higher image resolutions.

2) Training Strategy: The model is optimized by AdamW [42] with a learning rate of 1×10^{-5} for the backbone and 1×10^{-4} for the remainder of the architecture. Following the conventional protocols, the training data include the training splits of TrackingNet [36], LaSOT [38], and GOT-10k [39] (the prescriptive 1k sequences are removed to comply with the VOT2020 evaluation protocol [43]) and COCO2017 [44]. Since GOT-10k is a one-shot tracking benchmark, an additional model is trained for it only using the training split of it. The number of training epochs is 500 (200 for GOT-10k), each of which has 60k training instances. The learning rate decreases by 1/10 when the number of epochs reaches 400 (150 for GOT-10k), and then the validation process begins where the validation split of GOT-10k is used. We select the weight that yields the best validation performance among these 100 (50 for GOT-10k) candidates for evaluation purposes.

TABLE III
QUANTITATIVE RESULTS ON FOUR MAIN BENCHMARKS

Methods	GOT-10k*			TrackingNet			LaSOT			$\text{LaSOT}_{\text{ext}}$			Avg. Rank
	AO(%)	$\text{SR}_{0.5}$(%)	$\text{SR}_{0.75}$(%)	AUC(%)	P_{Norm}(%)	P(%)	AUC(%)	P_{Norm}(%)	P(%)	AUC(%)	P_{Norm}(%)	P(%)	
OTETrack_{256}	76.4	85.4	75.1	84.8	89.3	83.9	73.9	83.5	82.3	51.9	62.3	59.2	1.2
ARTrackV2_{256}	75.9	85.4	72.7	84.9	89.3	84.5	71.6	80.2	77.2	50.8	61.9	57.7	2.2
AQATrack_{256}	73.8	83.2	72.1	83.8	88.6	83.1	71.4	81.9	78.6	51.2	62.2	58.9	3.4
EVPTrack_{224}	73.3	83.6	70.7	83.5	88.3	82.4	70.4	80.9	77.2	48.7	59.5	55.1	5.6
ARTrack_{256}	73.5	82.2	70.9	84.2	88.7	83.5	70.4	79.5	76.6	46.4	56.5	52.3	6.7
SeqTrack-B_{256}	74.7	84.7	71.8	83.3	88.3	82.2	69.9	79.7	76.3	49.5	60.8	56.3	5.6
GRM_{256}	73.4	82.9	70.4	84.0	88.7	83.3	69.9	79.3	75.8	47.9	59.1	54.1	6.7
ROMTrack_{256}	72.9	82.9	70.2	83.6	88.4	82.7	69.3	78.8	75.6	48.9	59.3	55.0	7.4
TATrack-B_{256}	73.0	83.3	68.5	83.5	88.3	81.8	69.4	78.2	74.1	44.9	50.1	55.0	9.2
$\text{MixFormerV2-B}_{288}$	-	-	-	83.4	88.1	81.6	70.6	80.8	76.2	48.5	58.6	54.5	-
MixCvT_{320}	72.6	82.2	68.8	83.1	88.1	81.6	69.2	78.7	74.7	50.5	60.9	56.6	8.5
OSTrack_{256}	71.0	80.4	68.2	83.1	87.8	82.0	69.1	78.7	75.2	47.4	57.3	53.3	10.6
AiATrack_{320}	69.6	80.0	63.2	82.7	87.8	80.4	69.0	79.4	73.8	48.8	59.5	54.7	10.5
SwinTrack-T_{224}	71.3	81.9	64.5	81.1	85.8	78.4	67.2	76.4	70.8	47.6	58.3	53.9	12.1

'$*$' means that all results in this column are obtained via trained merely with GOT-10k.
'-' indicates that the corresponding weight is not provided so that the result cannot be attained.

3) Hyperparameters: The search image region and the template are acquired by extending the target bounding box by factors of 4 and 2, respectively. The hidden dimensionality of the decoder is 256. The vocabulary size, i.e., n_{bins}, is 4000. During the inference phase, the LH window size is 6000, and the peak area is one std on each side ($\beta = 1$). The dynamic template is updated if the interval reaches a certain value ($\gamma = 25$) and the matching score S is larger than a certain threshold ($\eta = 0.6$). The same threshold η is used for determining whether to use SES or HES for prediction purposes. The smoothing parameter a in SES is 0.9. The smoothing parameters b_1 and b_2 in HES are 0.7 and 0.8, respectively. The trajectory length (h) is set as 7.

4) Evaluation Metrics: The evaluation metrics are the default measures set by the data providers, i.e., the area under the curve (AUC; %), P_{Norm}(%) and P(%) for all datasets except GOT-10k, which employs the special evaluation metrics AO(%), $\text{SR}_{0.5}$ (%) and $\text{SR}_{0.75}$(%). The FPS is used to evaluate the efficiency of the tested models.

5) Platform: All experiments are conducted on two NVIDIA GeForce RTX 4090 GPUs. The source code is implemented in Python 3.9 and PyTorch 1.11.

C. Quantitative Results

The quantitative results obtained on four prevailing benchmarks are shown in Table III, and the average ranks of different trackers are given in the last column. The best, second-best, and third-best results produced in terms of each evaluation metric are highlighted in red, blue, and green, respectively. Our proposed OTETrack approach achieves the best general performance (Avg. Rank = 1.2). The analysis of the results is as follows.

1) Results Obtained on TrackingNet: TrackingNet is an immense dataset covering a plethora of real-world tracking scenes and various object classes. OTETrack_{256} is evaluated on its test split, and the results are submitted to the official evaluation server. The performance of OTETrack_{256} is second only to that of ARTrackV2_{256}, which is both a trajectory-based and feature-based method, as shown in Table III. However, the performance disparity between OTETrack_{256} and ARTrackV2_{256} is minor, which indicates that OTETrack_{256} can achieve state-of-the-art performance on TrackingNet without leveraging multiple historical image features. Moreover, the tracking tasks in TrackingNet are relatively easier than other benchmarks, which can be verified by the fact that the general AUC performances achieved on TrackingNet by all trackers are better than those attained on the other benchmarks. However, our proposed methods are tailored for the complicated tracking scenarios that require strong template matching and robust trajectory prediction capability. Therefore, it is not surprising that OTETrack_{256} fails to outperform ARTrackV2_{256} on TrackingNet but surpasses it on the other benchmarks.

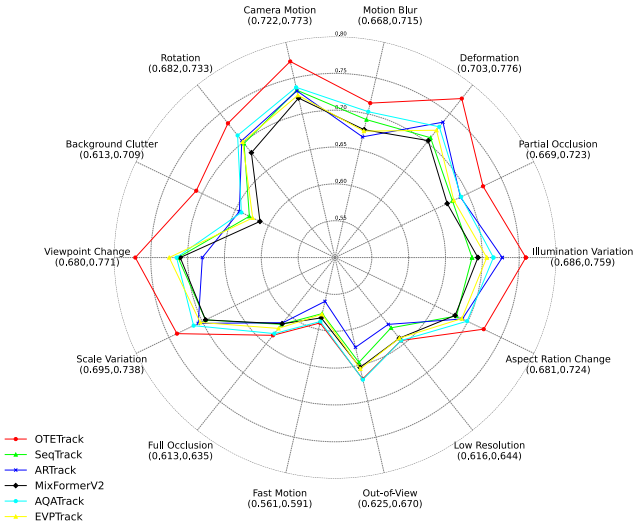
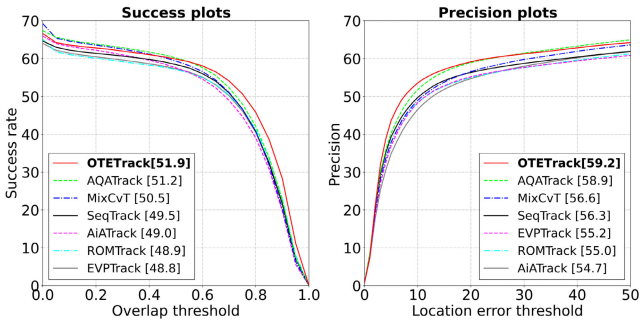


Fig. 6. The attribute-based evaluation of top-6 baselines on LaSOT (Except ARTrackV2 due to the absence of raw result). The worse and the best result of each attribute is given under the attribute name.

2) Results Obtained on LaSOT and LaSOT_{ext}: Both LaSOT and $\text{LaSOT}_{\text{ext}}$ are large-scale benchmarks with long-term video sequences. LaSOT contains 70 categories, and each category has four video sequences in the test split, while $\text{LaSOT}_{\text{ext}}$ provides an additional 150 video sequences belonging to 15 new categories. Specifically, the target objects in the many video sequences of $\text{LaSOT}_{\text{ext}}$ are small and move quickly. This poses great challenges for precisely performing visual tracking.

OTETrack_{256} achieves the best performance in terms of all three evaluation metrics on LaSOT and surpasses the



(a) (b)

Fig. 7. (a) The success plots where the tracking success is determined by the overlap of tracking bounding box and ground truth. Smaller overlap threshold leads to better tracking result but lower tracking quality. (b) The precision plots where the success is determined by the distance of the tracking bound box center and the ground truth center. The precision is calculated by the success ratio of all instances. Larger location error threshold results in better tracking result but lower tracking quality.

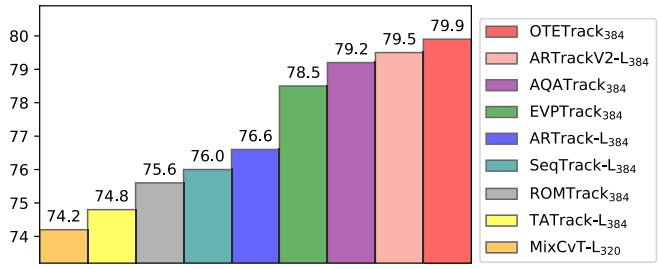


Fig. 8. The tracking performances of nine model variants, which possess larger image resolution or heavier model structure than those in Table III, on GOT-10k.

second-best ARTrackV2$_{256}$ method by 2.3%, 3.3%, and 5.1% with respect to the AUC, P_{Norm} and P measures, respectively. The elaborate attribute-based performances shown in Fig. 6 vividly explain why OTETrack$_{256}$ is able to achieve the best performance. Benefiting from the Overlapped ViT, OTETrack$_{256}$ is competitive in terms of handling problems involving image feature representation, so that it greatly outperforms other baselines under scenarios containing illumination variations, deformation, rotation, etc. OTETrack$_{256}$ also excels at handling scenarios with background clutter and partial occlusion due to its appropriate temporal information application strategy. The advantages of OTETrack$_{256}$ are marginal in cases with full occlusion, fast motion, low resolutions and out-of-view settings, where tracking failures are nearly unavoidable. In these cases, the abovementioned error accumulation problem frequently occurs such that the other trajectory-based models, e.g., ARTrack$_{256}$, have very poor performances, and the feature-based models, e.g., AQATrack$_{256}$, gain more resistance as the image information is ampler. However, the trajectory-based OTETrack$_{256}$ method still maintains its leading position, albeit not significantly, and outperforms the other trackers under nearly all four of these scenarios. This demonstrates the rationality of mildly applying trajectory prediction in the LH window.

OTETrack$_{256}$ maintains its leading position on LaSOT$_{ext}$ and achieves an AUC improvement of 0.7% over that of the second-best AQATrack$_{256}$ method. Fig. 7 further depicts the success plots and the precision plots yielded by the top 7 models on LaSOT$_{ext}$ to more comprehensively compare their tracking accuracies. By employing multiple historical image features, AQATrack$_{256}$ achieves competitive tracking performances, as shown in Fig. 7. However, the trajectory-based OTETrack$_{256}$ still outperforms the feature-based AQATrack$_{256}$, which indicates that the Overlapped ViT backbone is able to provide stronger spatial features. This in turn enables OTETrack$_{256}$ to achieve state-of-the-art tracking performances without relying on historical image features and implies that the temporal information application method in OTETrack$_{256}$ is robust enough to address difficult tracking circumstances. Specifically, AQATrack$_{256}$ can outperform OTETrack$_{256}$ only when the overlap threshold is low in the success plot or when the location error threshold is high in the precision plot, which indicates that the tracking quality of OTETrack$_{256}$ is higher than that of AQATrack$_{256}$.

3) Results on Obtained GOT-10k: GOT-10k is a special large-scale benchmark in which the target classes of the training set and the test set are completely different. This indicates that the template matching ability of a method is much more significant than its ability to extract the inner features of the search image and templates because the image features contained in the training set do not provide any information about those in the test set. As shown in Table III, OTETrack$_{256}$ achieves a new state-of-the-art AUC, which reaches 76.4%. Although ARTrackV2$_{256}$ achieves an identical performance to that of OTETrack$_{256}$ in terms of $SR_{0.5}$, its $SR_{0.75}$ performance is much worse than that of OTETrack$_{256}$ (72.7% vs. 75.1%). This indicates that the tracking quality of OTETrack$_{256}$ is extremely high because $SR_{0.75}$ requires a 0.25 larger intersection over union (IoU) between the tracking result and the ground truth than does $SR_{0.5}$ for gauging the tracking success. To validate the notion that the OTETrack model is still competitive when larger image resolutions are applied, we enlarge the image resolution from 256×256 to 384×384 to form the OTETrack$_{384}$ model variant. We then compare its performance with that of the heavier model variants of other baselines on GOT-10k. It can be observed that OTETrack$_{384}$ maintains its leading position and achieves an AUC of 79.9%, which is better than that of any other baseline in Fig. 8.

D. Efficiency Analysis

Benefiting from the efficient ANA module in the Overlapped ViT and the inexpensive temporal information application method, the success of OTETrack$_{256}$ is not built upon a heavier model. To validate that our proposed OTETrack method achieves both state-of-the-art accuracy and efficiency, we evaluate the comprehensive performances achieved by OTETrack$_{256}$ and other trackers on UAV123, which is an aerial visual tracking benchmark with 123 video sequences. Accuracy is measured by the AUC (%), and efficiency is measured by the FPS. As depicted in Fig. 9, our OTETrack$_{256}$ achieves a 70.8% AUC and 162 FPS (tracking speed), outperforming all other trackers with analogous resolutions in terms of both accuracy and efficiency.

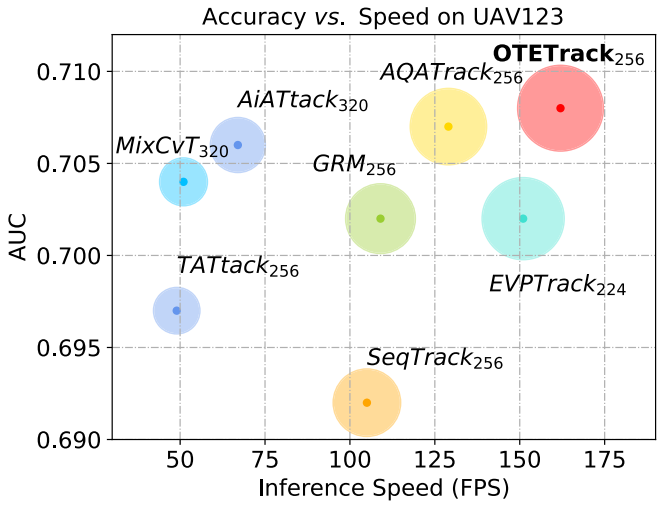


Fig. 9. The tracking performances, evaluated by both accuracy and speed, of eight trackers on UAV123. The radius of each tracker circle is computed by the multiplication of its AUC and FPS.

E. Ablation Study

We perform an ablation study to examine the individual functions of our proposed methods. The effectiveness of the spatial parts and the temporal parts are separately evaluated.

1) Effectiveness of the Spatial Parts: Five additional ablation variants of OTETrack$_{256}$ are tested.

1) *w/o OViT*: The Overlapped ViT backbone is replaced with the canonical ViT backbone.
2) *OTETrack$_{256}$-h*: The number of backbone layers is reduced to half of that contained in OTETrack$_{256}$ (from $12 \rightarrow 6$).
3) *w/o ANA*: The ANA modules are replaced with conventional attention modules, which perform attention to all of the patches simultaneously.
4) *w MP*: Instead of slicing the input image into O patches and A patches, 'w MP' slices the image into more overlapped patches by reducing the stride of the patch embedding module to half the patch size; thus, the total number of patches in the search image is $(16+15)^2 = 961$. The additional $(961-256) = 705$ patches are called A$'$ patches and are used in the ANA modules.

TrackingNet is chosen for the experiments since the spatial feature extraction ability of a model is more significant on such a large-scale benchmark where there are few complicated tracking circumstances. In such condition, the spatial tracking ability is more significant. The results are shown in Table IV. The floating-point operations (FLOPs) required for all the attention modules in the backbones of the five ViT-related models are also presented.

If the Overlapped ViT is replaced with the original ViT ('w/o OViT'), the performance degrades by a large margin (AUC: 84.8% $\rightarrow$ 83.7%), which indicates that feature extraction capabilities are of paramount significance for the spatial template matching process in deep visual trackers. This also demonstrates that the Overlapped ViT is truly able to enhance the spatial template matching ability of the basic ViT. The computational complexity of the attention process

TABLE IV
ABLATION STUDY ON THE EFFECTIVENESS OF THE SPATIAL PARTS

Models	TrackingNet			FLOPs
	AUC(%)	P_{Norm}(%)	P(%)	
OTETrack$_{256}$	84.8	89.3	83.9	5.0G
w/o OViT	83.7(-1.1)	88.4(-0.9)	82.8(-1.1)	2.7G
OTETrack$_{256}$-h	84.0(-0.8)	88.6(-0.7)	83.2(-0.7)	2.5G
w/o ANA	84.9(+0.1)	89.4(+0.1)	84.2(+0.3)	9.2G
w MP	85.1(+0.3)	89.5(+0.2)	84.3(+0.4)	20.6G

in the Overlapped ViT is also only double that of 'w/o OViT' (FLOPs: 5.0G vs. 2.7G), which is identical to the inference in Eq. (6). Specifically, OTETrack$_{256}$-h has half the number of backbone layers possessed by OTETrack$_{256}$, which means that its total computational complexity is close to that of 'w/o OViT'. However, OTETrack$_{256}$-h outperforms 'w/o OViT' in terms of the AUC by 0.3%, which demonstrates the efficiency of the Overlapped ViT.

If the ANA modules are replaced with conventional attention modules ('w/o ANA'), the computational complexity increases (FLOPs: 5.0G $\rightarrow$ 9.2G) to nearly four times the computational complexity of 'w/o OViT' (FLOPs: 9.2G vs. 2.7G) due to the quadratic complexity of the attention process. Although the inner feature extraction results obtained for the search image and templates are further reinforced by this approach, the tracking performance is only slightly enhanced (AUC: 84.8% $\rightarrow$ 84.9%). This indicates that the reinforcement to template matching is much more effective than the reinforcement to inner image feature extraction, which verifies the rationality of the ANA design.

If the images are sliced into more patches ('w MP'), the extent to which the complexity increases is even more severe than that exhibited by 'w/o ANA' (FLOPs: 9.2G $\rightarrow$ 20.6G), and the computational complexity reaches eight times that of 'w/o OViT' (FLOPs: 20.6G vs. 2.7G) since there are too many A$'$ patches. Although the performance enhancement (AUC: 84.8% $\rightarrow$ 85.1%) is slightly better than that provided by 'w MP', the computational cost is unacceptable. Hence, our method of choosing A patches achieves a good tradeoff between accuracy and efficiency.

2) Effectiveness of the Temporal Parts: We test four additional ablation variants with respect to the temporal parts of OTETrack$_{256}$ as follows.

1) *w/o ES*: The LH window simply takes the center of the previous target location as the window center instead of using the predicted location with ES.
2) *w/o HES*: SES is consistently used for predicting the LH window center regardless of the matching score threshold η, which means that the trend of the trajectory is always neglected.
3) *w/o SES*: HES is consistently used for predicting the LH window center regardless of the matching score threshold η, which means that the prediction method always assumes that a certain trend exists for an arbitrary trajectory.
4) *w/o LH*: The LH window is replaced with the conventional Hanning window, which means that trajectory

TABLE V
ABLATION STUDY ON THE EFFECTIVENESS OF THE TEMPORAL PARTS

Models	$\text{LaSOT}_{\text{ext}}$		
	AUC(%)	P_{Norm}(%)	P(%)
OTETrack_{256}	51.9	62.3	59.2
w/o ES	50.1(-1.8)	60.5(-1.8)	56.2(-3.0)
w/o HES	51.0(-0.9)	62.0(-0.3)	57.9(-1.3)
w/o SES	50.5(-1.4)	60.8(-1.5)	56.7(-2.5)
w/o LH	51.0(-0.9)	61.9(-0.4)	58.0(-1.2)

turbulence is no longer considered, and the window center is still predicted based on the trajectory.

$\text{LaSOT}_{\text{ext}}$ is chosen for the experiments because temporal information is more significant when the target objects are small and move quickly. The results are shown in Table V. The following can be observed.

If the LH window is not predicted with the trajectory ('w/o ES'), the achieved tracking performance is much worse than that of the original OTETrack_{256} model (AUC: 51.9% $\rightarrow$ 50.1%). This indicates that applying trajectory information to visual tracking is effective and that integrating it into the Hanning window is feasible.

If only the SES method is adopted for trajectory prediction ('w/o HES'), then the trajectory trend, which manifests the target motion, is neglected. Since the model is no longer able to exploit the trend information to capture more accurate target locations, the performance of 'w/o HES' decreases as expected (AUC: 51.9% $\rightarrow$ 51.0%). However, if only HES is adopted for trajectory prediction ('w/o SES'), then the model is prone to spurious trajectory information, similar to the majority of trajectory-based trackers. Once a tracking failure occurs, it affects the fitted trend term of HES in the upcoming tracking frames, and the extent of this impact can be considerable if the tracking failure is so far from the ground truth that even ES fails to counteract its influences. Therefore, the performance of 'w/o SES' is even worse than that of 'w/o HES' (AUC: 51.0% $\rightarrow$ 50.5%) on the complicated tracking instances of $\text{LaSOT}_{\text{ext}}$. Conclusively, it is essential for OTETrack_{256} to use the matching score threshold η to determine whether to use SES or HES.

If the conventional Hanning window is employed in OTETrack_{256} ('w/o LH'), the tracking performance decreases (AUC: 51.9% $\rightarrow$ 51.0%). Therefore, trajectory turbulence needs to be considered to mitigate the influence of spurious trajectory information, and its solution in OTETrack_{256}, i.e., the LH window, plays an important role.

F. Parameter Sensitivity

1) Matching Score Threshold: The matching score threshold η controls the selection of the appropriate ES forecasting method and the dynamic template updating process. Setting η too low causes unqualified spatial tracking results to affect the subsequent tracking step, whereas setting η too high makes the model unable to fully use the available temporal information. Therefore, it is necessary to evaluate the effects of different η values on the resulting tracking accuracy. GOT-10k is used as the evaluation dataset. It can be concluded from Fig. 10 that

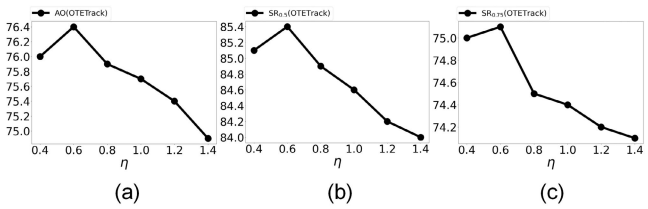


Fig. 10. The performances of OTETrack_{256} with different matching score thresholds η on GOT-10k. (a) AO vs. η. (b) $\text{SR}_{0.5}$ vs. η. (c) $\text{SR}_{0.75}$ vs. η.

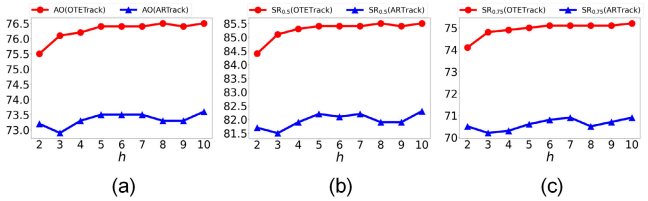


Fig. 11. The performances of OTETrack_{256} (red) and ARTrack_{256} (blue) with different trajectory lengths h on GOT-10k. (a) AO vs. h. (b) $\text{SR}_{0.5}$ vs. h. (c) $\text{SR}_{0.75}$ vs. h.

$\eta = 0.6$ produces the best performance; thus, it is adopted as the default setting for OTETrack_{256}.

2) Trajectory Length: We also evaluate the sensitivity of the performance achieved by OTETrack_{256} to the trajectory length h. The empirical results of many other works have shown that their methods are only capable of handling short-term trajectories and suffer from performance degradation when the trajectory length is too long. We use ARTrack_{256} for comparison purposes and adopt GOT-10k for the experiment since ARTrack_{256} is also a trajectory-based method that leverages historical trajectories for prediction. As shown in Fig. 11, OTETrack_{256} achieves better tracking results when h is prolonged, and its performance converges with the last few large values of h due to the ES mechanism. However, the performance of ARTrack_{256} is turbulent, and longer trajectory durations do not always lead to its better tracking results, which illustrates that the parametric prediction method is unstable and unreliable for nonstationary trajectory prediction tasks. It can also be observed from Fig. 11 that OTETrack_{256} already approaches the best tracking performance when $h = 7$, so we take $h = 7$ as our default setting for model efficiency.

G. Case Study

We present visualizations to vividly distinguish between the performances of OTETrack_{256} and its competitors in terms of both tracking results and latent feature representations. We select the difficult tracking situations in $\text{LaSOT}_{\text{ext}}$, visualize the matching score distributions produced during detokenization in several typical cases via heatmaps and depict the entire tracking trajectories to holistically portray the tracking ability of OTETrack_{256}. ARTrack_{256} is adopted for comparison purposes, as it is also a one-stream visual tracker with trajectory-based temporal information application and tokenization techniques. As shown in Fig. 12 and Fig. 13, both ARTrack_{256} and OTETrack_{256} successfully locate the target object at the beginning. However, when the background clutter problem occurs in Figs. 12(b3)(b5) and the out-of-view

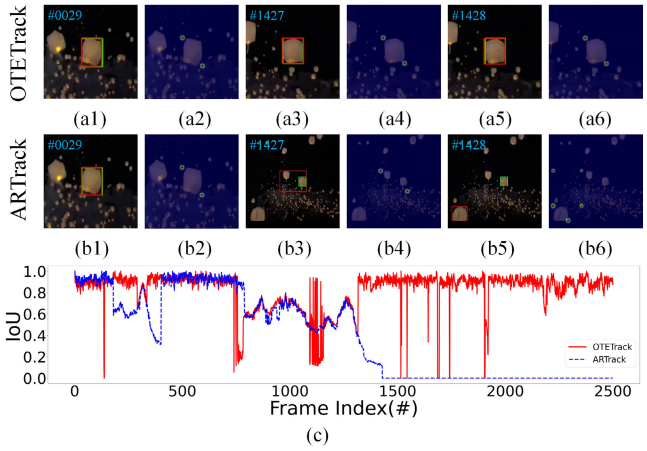


Fig. 12. Visualization of OTETrack and ARTrack on lantern-7 of $\text{LaSOT}_{\text{ext}}$. The frame indexes are in the frames. (a) and (b) are respectively the result of OTETrack and ARTrack. The subfigures with odd indexes (1, 3, 5) are tangible frames and the subfigures with even indexes (2, 4, 6) are the corresponding matching score heat maps during detokenization. The green rectangles are the ground truths and the red rectangles are the tracking results. The curves in (c) plot frame index versus IoU.

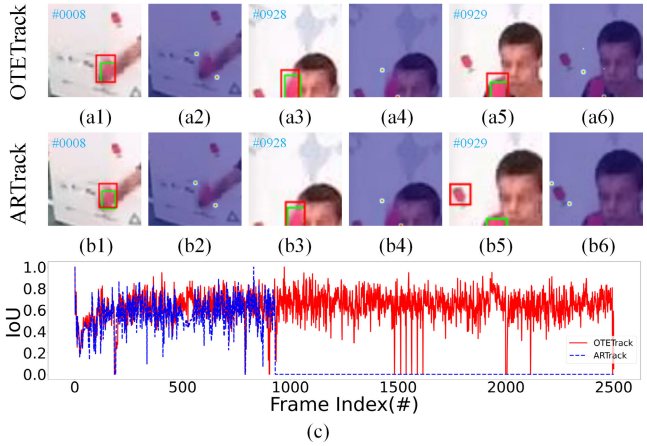


Fig. 13. Visualization of OTETrack and ARTrack on jianzi-1 of $\text{LaSOT}_{\text{ext}}$. Other notation usages are identical with those in Figure 12.

problem occurs in Figs. 13(b3)(b5), ARTrack_{256} loses the target and starts to trace other similar objects. In contrast, OTETrack_{256} consistently locks onto the target, which implies that OTETrack_{256} is better at handling complicated tracking conditions. The heatmaps of the matching score distributions also show the superior stability of OTETrack_{256}, especially compared with that of ARTrack_{256}, as shown in Fig. 12(a6) and Fig. 12(b6). Moreover, the IoU trajectories show that ARTrack_{256} eternally loses the target if a tracking failure occurs (IoU $=$ 0) due to multiple factors. However, the general tracking process of OTETrack is not interrupted by several transient tracking failures, which indicates that the temporal information application strategy of OTETrack_{256} is more robust to spurious trajectory information.

VI. Conclusion and Future Work

This work proposes OTETrack, a novel tracker that achieves state-of-the-art single-object tracking performance due to its Overlapped ViT backbone and robust temporal information application strategy. To be precise, the proposed Overlapped ViT is an ad hoc backbone tailored for one-stream visual tracking. It slices images into overlapped patches to enhance the spatial template matching ability of OTETrack and maintains the efficiency with inner ANA modules. Moreover, the temporal information applied in OTETrack is materialized by integrating trajectory prediction into the LH window, which ensures that the temporal information affects the spatial tracking results in a mild manner. Thus, OTETrack is resistant to tracking failures and spurious temporal information. Moreover, the spatial tracking quality of ES-based trajectory prediction is used to determine the appropriate form of the ES method, thus further enhancing the robustness of OTETrack. Extensive experiments conducted on five benchmarks with thirteen baselines verify the state-of-the-art tracking ability of OTETrack.

In the future, we will continue to delve into the realm of visual tracking. Furthermore, the additional A patches contained in the Overlapped ViT proposed in this work are not adaptively chosen, which means that there is space to further reduce the additional number of patches required for attaining better efficiency. Moreover, adaptivity is also a problem in our temporal information application strategy, as the matching score threshold needs to be empirically and manually chosen to achieve the best performance. Besides, OTETrack does not show apparent advantage on tackling the tracking scenarios involving full occlusion, fast motion, low resolution and out-of-view. We plan to resort to reinforcement learning to solve the above problems in the future.

References

[1] B. Sun, Z. Wang, S. Wang, Y. Cheng, and J. Ning, "Bidirectional interaction of CNN and transformer feature for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 13, 2024, doi: 10.1109/TCSVT.2024.3376690.

[2] Y. Liang, H. Chen, Q. Wu, C. Xia, and J. Li, "Joint spatio-temporal similarity and discrimination learning for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 13, 2024, doi: 10.1109/TCSVT.2024.3377379.

[3] D. Zhang, X. Xiao, Z. Zheng, Y. Jiang, and Y. Yang, "Probabilistic assignment with decoupled IoU prediction for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 7, pp. 5776–5789, Jul. 2024.

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 850–865.

[5] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 8126–8135.

[6] 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. Eur. Conf. Comput. Vis.*, 2022, pp. 341–357.

[7] S. Gao, C. Zhou, C. Ma, X. Wang, and J. Yuan, "AiATrack: Attention in attention for transformer visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 146–164.

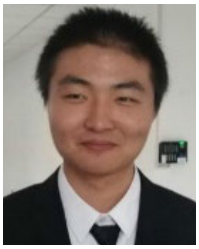
[8] X. Chen, H. Peng, D. Wang, H. Lu, and H. Hu, "SeqTrack: Sequence to sequence learning for visual object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 14572–14581.

[9] X. Wei, Y. Bai, Y. Zheng, D. Shi, and Y. Gong, "Autoregressive visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9697–9706.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[11] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–22.

[12] N. Bourriez et al., "ChAda-ViT: Channel adaptive attention for joint representation learning of heterogeneous microscopy images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 11556–11565.
[13] I. Ntinou, E. Sanchez, and G. Tzimiropoulos, "Multiscale vision transformers meet bipartite matching for efficient single-stage action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18827–18836.
[14] 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.
[15] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2021, pp. 9992–10002.
[16] L. Lin, H. Fan, Z. Zhang, Y. Xu, and H. Ling, "Swintrack: A simple and strong baseline for transformer tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 16743–16754.
[17] L. Shi, B. Zhong, Q. Liang, N. Li, S. Zhang, and X. Li, "Explicit visual prompts for visual object tracking," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 4838–4846.
[18] J. Xie et al., "Autoregressive queries for adaptive tracking with spatio-temporaltransformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 19300–19309.
[19] K. He, C. Zhang, S. Xie, Z. Li, and Z. Wang, "Target-aware tracking with long-term context attention," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 773–780.
[20] E. S. Gardner, "Exponential smoothing: The state of the art," *J. Forecasting*, vol. 4, no. 1, pp. 1–28, Jan. 1985.
[21] 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.
[22] Y. Cui, T. Song, G. Wu, and L. Wang, "MixFormerV2: Efficient fully transformer tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 58736–58751.
[23] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.
[24] X. Yuan et al., "Multi-step temporal modeling for UAV tracking," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 11, 2024, doi: 10.1109/TCSVT.2024.3375366.
[25] Y. Cai, J. Liu, J. Tang, and G. Wu, "Robust object modeling for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2023, pp. 9589–9600.
[26] T. Chen, S. Saxena, L. Li, D. J. Fleet, and G. Hinton, "Pix2seq: A language modeling framework for object detection," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–17.
[27] Y. Bai, Z. Zhao, Y. Gong, and X. Wei, "ARTrackV2: Prompting autoregressive tracker where to look and how to describe," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 19048–19057.
[28] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10448–10457.
[29] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.
[30] Y. Xu et al., "FDViT: Improve the hierarchical architecture of vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 5927–5937.
[31] S. Wu, T. Wu, H. Tan, and G. Guo, "Pale transformer: A general vision transformer backbone with pale-shaped attention," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 2731–2739.
[32] Z. Tu et al., "MaxViT: Multi-axis vision transformer," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 459–479.
[33] C. Zheng, Y. Zhang, J. Gu, Y. Zhang, L. Kong, and X. Yuan, "Cross aggregation transformer for image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, Oct. 2022, pp. 25478–25490.
[34] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
[35] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
[36] 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)*, Jun. 2018, pp. 300–317.
[37] 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. 5374–5383.
[38] H. Fan et al., "LaSOT: A high-quality large-scale single object tracking benchmark," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 439–461, Feb. 2021.
[39] 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.
[40] 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.
[41] X. Zhang et al., "HiViT: A simpler and more efficient design of hierarchical vision transformer," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–15.
[42] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19.
[43] M. Kristan et al., "The eighth visual object tracking VOT2020 challenge results," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 547–601.
[44] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.



Li Shen received the B.S. degree in navigation, guidance and control from the School of Automation Science and Electrical Engineering, Beihang University, China. He is currently pursuing the Ph.D. degree in navigation, guidance and control. His research interests include computer vision and time series analysis.

Xuyi Fan received the B.S. degree in unmanned aerial vehicle system engineering from the Flying College, Beihang University, China. He is currently completing a M.A.Eng. degree in electronic information engineering with the School of Electronics and Information Engineering. His research interests include deep learning and computer vision.

Hongguang Li is currently a Senior Engineer with Beihang University. His main research interests include intelligent optical image processing and end-to-side computing applications for unmanned systems, including optical image restoration and enhancement, target detection and tracking, geometric correction and target positioning, end-to-side computing software, and hardware system design and application.