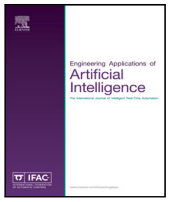
Engineering Applications of Artificial Intelligence 127 (2024) 107304

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



TATrack: Target-aware transformer for object tracking

Kai Huang [a], Jun Chu [a,*], Lu Leng [a], Xingbo Dong [b]

[a] Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang, 330063, China
[b] School of Artificial Intelligence, Anhui University, Anhui, 230093, China

ARTICLE INFO

Keywords:
Object tracking
Siamese visual tracker
Self-attention
Deformable attention
Target-aware tracking

ABSTRACT

Vision transformers have recently been adapted for object tracking and achieved promising performances owing to global correlation modeling using a self-attention mechanism. However, self-attention in existing trackers pays equal attention to the foreground and background, leading to a limited discriminative ability because attention is not target-aware. Existing solutions suffer from issues associated with information loss and the introduction of additional noise. This study proposes a Transformer-based Siamese tracking architecture integrated with deformable attention called TATrack. The TATrack can focus on the most relevant information about the target in the search region by adaptively selecting the positions of the key and value pairs, thereby reducing the information loss and additional noise. Experiments demonstrate that TATrack outperforms state-of-the-art models by a significant margin on GOT-10k, TrackingNet, LaSOT, and OTB100, with comparable processing speeds. The source code and pretrained models are available at www.github.com/Kevoen/TATrack.

1. Introduction

Visual object tracking is an important task in computer vision. Given a target in the initial frame, object tracking aims to consistently estimate the target position in subsequent video frames. Numerous technologies, including autonomous driving (Javed et al., 2022b), video surveillance (Javed et al., 2022a), human–computer interaction (Ma'arif et al., 2021; Bagherzadeh and Toosizadeh, 2022; Kurdthongmee et al., 2022), and virtual reality (Jiao et al., 2021), have been used for object tracking. Although significant progress has been made recently, practical applications remain limited owing to issues such as background distractions, interference from similar objects, and object appearance variations.

Recently, using the transformer architecture (Vaswani et al., 2017; Yang et al., 2022), state-of-the-art performances have been achieved in classification and detection vision tasks in various studies, such as TransT (Chen et al., 2021), DTT (Yu et al., 2021), CSWinTT (Song et al., 2022), and STARK (Yan et al., 2021). The success of transformers in high-level visual tasks is primarily because of the strong representation power achieved by the self-attention mechanism with a large or even global receptive field.

A current research gap is that incorporating self-attention into object tracking algorithms directly can result in a performance decline. This is because the self-attention mechanism treats both the target and background in the search regions as equally important; consequently, the features can be negatively influenced by irrelevant elements that exist outside the region of interest (Chen et al., 2021; Yu et al., 2021; Yan et al., 2021) (Fig. 1). In object tracking, the tracker in the search region is expected to focus on the information most relevant to the target and suppress background distractions. Trackers using self-attention typically suffer from a lack of focus on the most relevant information in the search regions. Therefore, this study aims to enhance the attention of self-attention based trackers towards the most relevant information in the search region, thereby improving the discrimination capability and tracking accuracy.

Several transformer-based trackers have been proposed to address these issues. As each pixel value of the attention features is calculated by all pixel values of the input features in the naive self-attention, which blurs the edge regions of foreground, SparseTT (Fu et al., 2022) employs a sparse multi-head self-attention (SMSA) to improve the discrimination of the foreground-background and alleviate the ambiguity of the edge regions of the foreground. The SMSA only determines each pixel value of the attention features using the K pixel values (top-K row of the similarity matrix) that are most similar to it, which makes the foreground more focused and the edge regions of the foreground more discriminative. Informer (Zhou et al., 2021) was originally proposed for time-serious data but shares similarities with SparseTT; it selects the most relevant information token from the similarity matrix in a row-by-row manner based on the Kullback–Leibler divergence.

Although these studies have achieved significant performance improvements, using arbitrary rows from the similarity matrix suffers

* Corresponding author.
E-mail address: chuj@nchu.edu.cn (J. Chu).

https://doi.org/10.1016/j.engappai.2023.107304
Received 24 April 2023; Received in revised form 20 August 2023; Accepted 12 October 2023
Available online 27 October 2023

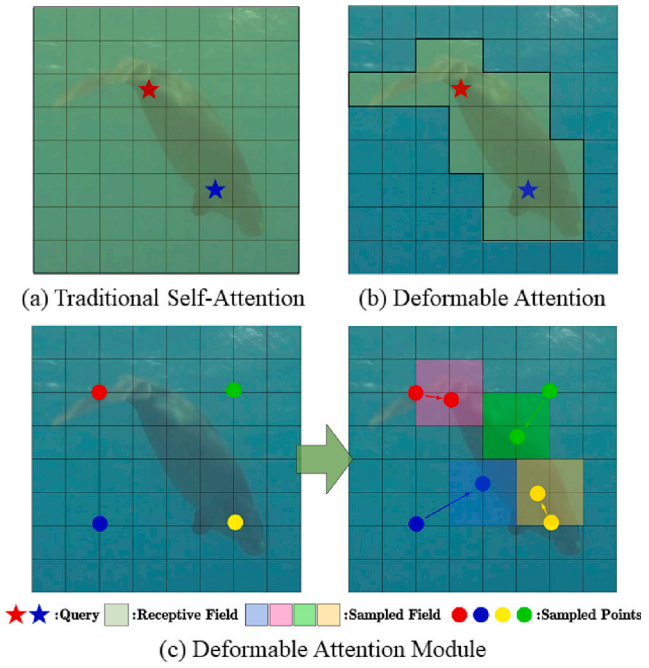


Fig. 1. Comparison of our deformable attention with traditional self-attention. Blue and red stars indicate different queries in the search region, and dark areas indicate receptive fields. (a) Traditional self-attention mechanisms focus on global information. (b) The deformable attention mechanism focuses on the most relevant information about the target. (c) The left figure shows the initial sampled points, and the right indicates the adaptively sampled fields on the offsets.

from two challenges. First, existing approaches suffer from unavoidable information loss by replacing the remaining rows of the similarity matrix with zeros, resulting in the under-utilization of the matrix. Second, an arbitrary selection of rows may introduce additional noise, thereby negatively affecting the performance of the trackers.

In summary, using self-attention directly and selecting arbitrary rows from the similarity matrix are suboptimal methods for the visual object-tracking task. Motivated by the deformable attention from Xia et al. (2022), we propose a target-aware tracker (TATrack) to address these issues. The TATrack is a transformer-based Siamese tracking architecture based on deformable attention for target-aware visual tracking. By avoiding manual feature manipulation, TATrack reduces information loss. In addition, the deformable attention mechanism adaptively samples the most relevant information about the target in the search region to reduce additional noise, thereby paying more attention to the foreground information and improving the model's tracking performance.

The following is a summary of the main contributions of this study:

- We propose the TATrack, a transformer-based Siamese tracking architecture that is target-aware and capable of accurately tracking various complex scenes, such as illumination variation, partial or full occlusion, and background clutter.
- We integrate deformable attention into the TATrack to achieve target-aware attention, thereby improving the differentiation of the target and background in search regions and avoidance of background distractions.
- Extensive experiments and analyses were performed on six public benchmark datasets: OTB100, LaSOT, UAV123, NFS30, TrackingNet, and GOT-10K. The results demonstrate the superior performance of the TATrack.

The remainder of this article is organized as follows. Section 2 briefly introduces Siamese-based and Transformer-based trackers. Section 3 discusses the basic conceptS of self-attention and deformable attention. Section 4 describes the proposed end-to-end target-aware transformer-based tracker with deformable attention. Section 5 presents the experiments and results. Finally, Section 6 concludes the paper.

2. Related work

2.1. Siamese based tracking

The Siame-based object tracker treats the object- tracking task as a template-matching task. SiamFC (Bertinetto et al., 2016) first introduced the Siamese framework and achieved a remarkable performance improvement. SiamRPN (Li et al., 2018) implemented a region proposal network (RPN) on top of SiamFC to address multiscale variations and improve the performances of Siamese-based trackers. However, SiamRPN cannot use deeper backbone networks; therefore, SiamRPN++(Li et al., 2019) and SiamDW (Zhang et al., 2019) were proposed to address this problem. SiamRPN++ (Li et al., 2019) adopts ResNet as the backbone network, significantly improving the tracking accuracy and robustness. In contrast, SiamDW (Zhang et al., 2019) uses a deeper and wider backbone network structure to improve the tracking performance.

Because the output of the aforementioned anchor-based tracker is a bounding box, the accuracy of the target position prediction is naturally upper bounded by the bounding box. Anchor-based trackers are sensitive to hyperparameters, such as the number of anchors, scale ratio, and aspect ratio. Recently, anchor-free trackers have attracted significant attention. SiamMask (Wang et al., 2019) improved the tracking accuracy by integrating segmentation tasks into the tracking. SiamFC++ (Xu et al., 2020) added a quality evaluation score branch parallel to the classification and regression branch to improve the tracking accuracy. To perform the classification, SiamCAR (Guo et al., 2020) used a center-ness branch parallel to the classification branch of the prediction head. Ocean (Zhang et al., 2020) employed an anchor-free object-tracking network based on feature alignment, inspired by the FCOS (Tian et al., 2019) originally designed for object detection.

2.2. Transformer in tracking

The transformer was originally proposed for natural language processing tasks in machine translation. DETR (Carion et al., 2020) first introduced a transformer to the object detection task and achieved significant performance improvement. Thereafter, transformers were increasingly used in computer vision applications, and various transformer-based networks have been proposed, such as ViT (Dosovitskiy et al., 2020) and Swin transformers (Liu et al., 2021).

Inspired by the success of transformers in vision tasks (Khan et al., 2022), researchers began to apply transformers in object tracking tasks, and the combination of convolutional neural network (CNN) and transformer paradigms has proven effective in tracking tasks. In TransT (Chen et al., 2021), the transformer was integrated into the Siamese framework to improve the tracking performance by enhancing and fusing features. STARK (Yan et al., 2021) employed an encoding–decoding transformer-based spatio-temporal architecture; the encoder learns to model the relationship between the template and search region, and the decoder learns to predict the target's position using the query embedding. DTT (Yu et al., 2021) used an encoder–decoder transformer structure instead of traditional discriminator modules, thereby improving the tracking performance. In addition to using the transformer to replace a part of the traditional tracking networks, InMo (Guo et al., 2022) adds branch-wise interactions to the transformer backbones. CSWinTT (Song et al., 2022) models the relationship between the target template and search region by leveraging the transformer architecture and multiscale cyclic shifting window attention.

Several studies have been conducted using only transformers for object tracking. SwinTrack (Lin et al., 2021) utilizes a transformer for representation learning and feature fusion to achieve a significant

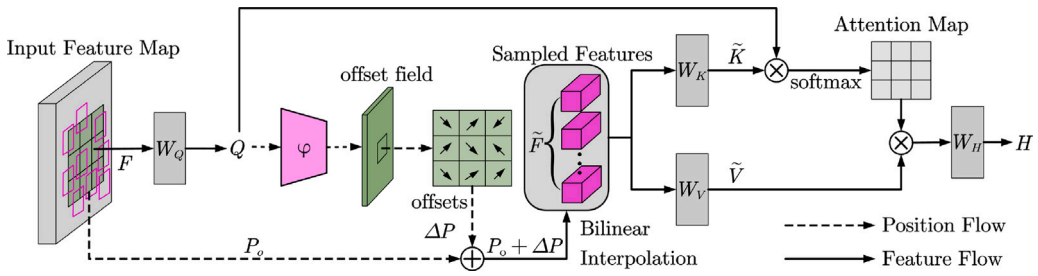


Fig. 2. Illustration of the deformable attention mechanism. Unlike the traditional self-attention mechanism, deformable attention uses a multi-layer perception network φ to learn the offsets ΔP of the reference points P_o in the search region and then uses bilinear interpolation to sample the features, which are then mapped into key–value pairs by linear projection. Finally, Q, $\tilde{K}$, and $\tilde{V}$ are processed using conventional self-attention mechanism operations.

performance improvement. SparseTT (Fu et al., 2022) is a transformer Siamese network framework that employs a sparse transformer with an encoding–decoding structure to focus on the most relevant information in the search region. Recently, Ye et al. (2022), Chen et al. (2022) and Cui et al. (2022) proposed an end-to-end transformer-based framework for object tracking.

Although the self-attention mechanism (Vaswani et al., 2017) can model pixel-to-pixel correlations to improve the model's long-range modeling capability, gaps remain when directly applying self-attention to object tracking. Object tracking differs from other tasks (such as classification and detection) because only one target is typically tracked in object tracking; therefore, the system should focus on the target and avoid distractions from other objects. Unfortunately, the attention of the transformer is equalized, which is a problem in object-tracking scenarios. Hence, we propose a transformer-based Siamese tracking architecture based on deformable attention.

3. Preliminaries

3.1. Multi-head self-attention

Given input $x \in \mathbb{R}^{C\times H\times W}$ in conventional multi-head self-attention (MHSA), queries Q_i, keys K_i, and values V_i can be computed as:

$$Q_i = xW_Q, K_i = xW_K, V_i = xW_V, \tag{1}$$

where $Q_i, K_i, V_i \in \mathbb{R}^{HW\times C}, i = 1, \dots, h$, and h is the number of heads. Therefore, the MHSA operation is defined as:

$$MHSA(Q, K, V) = Concat(H_1, \dots, H_h)W_H, \tag{2}$$

where Concat is the concatenation operation, and H_i is the attention vector of the output of the ith head, computed as follows:

$$H_i = Softmax(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i, \tag{3}$$

where d_k is the dimension of the key, and $W_Q, W_K, W_V, W_H \in \mathbb{R}^{C\times C}$ is the projection matrix.

In self-attention, the similarity matrix $M = \frac{QK^T}{\sqrt{d_k}}$ of each query-key pair is computed independently. Therefore, the foreground and background features are equally focused, making the discrimination of the target from the background difficult.

3.2. Deformable attention

For most vision tasks, the candidate key–value set for a given query is typically expected to be adaptable and flexible, with the capacity to adapt to each distinct input. Deformable attention for transformers (Xia et al., 2022) is a recent approach that can achieve these goals. Inspired by the deformable attention in Xia et al. (2022), we propose integrating the deformable attention into a transformer-based Siamese architecture for object tracking, thereby allowing the candidate key–value set for a given query to be flexible.

Fig. 2 presents a general overview of deformable attention (Xia et al., 2022). In particular, the focused regions are first determined using numerous groups of deformed sampling points predicted by an offset network with queries as inputs. Using the important regions of the feature maps, deformable attention can effectively model the relationships between tokens. Subsequently, bilinear interpolation employs sample features from the feature maps and feeds these features into key and value projections to build deformed keys and values. Finally, conventional multi-head attention is applied to attend to the sampled keys and aggregate features from the deformed values.

Formally, given an input feature map $F \in \mathbb{R}^{H\times W\times C}$, the initial sampling point is defined as a regular grid $P_o \in \mathbb{R}^{H_g\times W_g\times 2}$, where the size of the sampled grid is $H_g = H/c, W_g = W/c$, and c is the scale factor of the downsampled input features. According to deformable convolutional networks (DCN), normalizing the initial sampling point in the range $[-1, 1]$ can be defined as:

$$\begin{aligned} P_o &= Normal(\{(0,0),(0,1),\dots,(H_g-1, W_g-1)\}) \\ &= \{(-1,-1),\dots,(1,1)\}. \end{aligned} \tag{4}$$

To obtain the offset at each initial position, input feature F is linearly mapped into query $Q = FW_Q$, which is then fed into a multi-layer perceptron (MLP) network $\varphi(\cdot)$ to generate the deviation Δp_n for the corresponding position, which consists of two nonlinear convolutional layers. The input features were first captured using a 5×5 deep convolution to capture the local features. Subsequently, Gaussian error linear unit (GELU) activation and 1×1 convolution are used to obtain the two-dimensional offsets. ΔP is defined as follows:

$$\Delta P = \varphi(FW_Q), \tag{5}$$

where $\Delta P = \{\Delta p_n | n = 1, \dots, N\}$, and $N = |P_o|$ is the number of sampling points. The following operations are performed on input feature map F:

$$\begin{aligned} &\tilde{F} = \theta(F; P_o + \Delta P) \\ &Q = FW_Q, \tilde{K} = \tilde{F}W_K, \tilde{V} = \tilde{F}W_V, \end{aligned} \tag{6}$$

where $\tilde{F}$ denotes the sampled features; $\tilde{K}$ and $\tilde{V}$ are the deformed keys and values, respectively; and $W_Q, W_K, W_V \in \mathbb{R}^{C\times C}$ is the projection matrix. $\theta(\cdot;\cdot)$ is denoted as a bilinear interpolation:

$$\theta(x; (p_x, p_y)) = \sum_{(q_x, q_y)} G(p_x, q_x)G(p_y, q_y)x[q_x, q_y, :], \tag{7}$$

where $G(a, b) = max(0, 1-|a-b|)$, and $x \in \mathbb{R}^{H\times W\times C}$ is indexed by (q_x, q_y) for all positions. Because $G(\cdot,\cdot)$ is non-zero only at the four integer points closest to (p_x, p_y), the formula is reduced to a weighted average of the four positions.

After obtaining the queries, keys, and values, similar to the multi-head attention mechanism, the multi-head deformable target-aware attention operation (TAMHSA) is defined as:

$$TAMHSA(Q, \tilde{K}, \tilde{V}) = Concat(H_1, \dots, H_h)W_H, \tag{8}$$

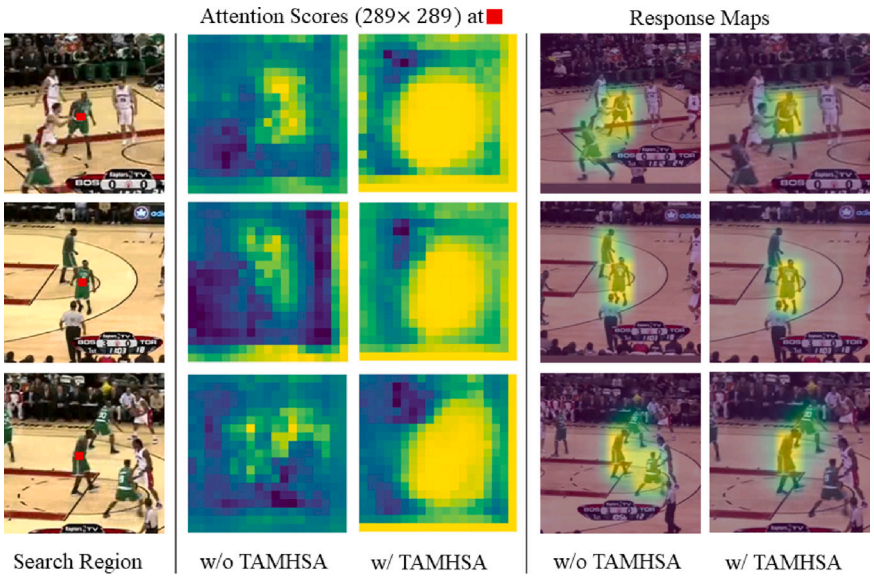


Fig. 3. Visualization of the attention scores and response maps. We visualized several attention score maps and response with and without using deformable attention.

where Concat is the concatenation operation, and H_i denotes the self-attention feature of the output of the ith head and can be computed as:

$$H_i = Softmax(\frac{Q_i \tilde{K}_i^T}{\sqrt{d_k}})\tilde{V}_i, \tag{9}$$

where d_k is the dimension of the key, and $W_H \in \mathbb{R}^{C\times C}$ is the projection matrix.

Fig. 3 compares the attention and response maps of a key in the transformer with and without deformable attention. Deformable attention can focus more on features related to the target and suppress distractions. Compared with traditional self-attention, deformable attention in object tracking offers several advantages. It improves the localization accuracy, enhances the robustness to scale and deformation, enables selective feature extraction, models long-range dependencies, and adapts to different object classes. These advantages contribute to a significantly more reliable, accurate, and adaptable object-tracking performance.

4. Method

We present TATrack, a target-aware deformable attention transformer for object tracking. As illustrated in Fig. 4, the TATrack is a Siamese architecture consisting of three components: a feature extraction module, target-aware transformer network based on deformable attention, and prediction head. The feature extraction module is a parameter-sharing Swin transformer backbone network. Target-aware transformer networks are encoding–decoding transformer networks that use a deformable attention mechanism. This allows the tracker to focus more on the target features, thereby enhancing the target information and suppressing background distraction. The prediction head distinguishes the foreground from the background and predicts the location of the target using a bounding box.

4.1. Feature extraction module

Recently, vision transformers have achieved excellent performances in computer vision tasks. Compared with their CNN counterparts, this transformer can extract more compact feature representations and richer semantic information; hence, better target location prediction performances can be achieved (Lin et al., 2021). Therefore, a Swin transformer was used as the backbone of the feature-extraction module.

As shown in Fig. 4, the TATrack input is an image pair, i.e., template image $x \in \mathbb{R}^{3\times H_x\times W_x}$ and search region image $z \in \mathbb{R}^{3\times H_z\times W_z}$. First, template images x and search region image z are split into non-overlapping patches using a patch splitting module, obtaining $x_p \in \mathbb{R}^{N_x\times(3\times P\times P)}$ and $z_p \in \mathbb{R}^{N_z\times(3\times P\times P)}$, where $P\times P$ is the resolution of each patch, and $N_x = H_xW_x/P^2, N_z = H_zW_z/P^2$ denote the numbers of patches of x and z, respectively. According to the tiny version of the Swin transformer (Swin-T), we slice the image pairs into 4 × 4 resolution patches, i.e., $P = 4$. Template token $X = \phi(x_p) \in \mathbb{R}^{\frac{H_x}{s}\frac{W_x}{s}\times C}$ and search region token $Z = \phi(z_p) \in \mathbb{R}^{\frac{H_x}{s}\frac{W_x}{s}\times C}$ are then generated, where $\phi(\cdot)$ is the Swin transformer backbone; s denotes the strides of the backbone network and C is the number of hidden channels in the model.

4.2. Target-aware deformable attention transformer

Given the token extracted by the feature extraction module, an encoder–decoder transformer based on deformable attention is proposed to build the correlation between the template and search region features. The encoder encodes the template features, and the decoder encodes the search region features and simultaneously builds the correlation between the search region feature and template. Finally, target-aware features are generated and passed to the double-head predictor for regression and classification. Fig. 4 overviews the deformable attention of the transformer.

Encoder. Similar to the structure of the encoder in a conventional encoder–decoder transformer, each encoder layer consists of an MHSA and feed-forward neural network (FFN). The FFN consists of a two-layer MLP and GELU activation layer. Layer normalization and residual concatenation were applied after each module (MHSA and FFN). The encoder consists of a stack of N layers, and the input of each layer is the output of the previous layer.

As shown in Fig. 4, a spatial location encoding $Pos_{enc} \in \mathbb{R}^{H_zW_z\times C}$ was added to template feature $Z \in \mathbb{R}^{H_zW_z\times C}$ and then fed to the encoder. The encoder process can be described as follows:

$$\begin{aligned} f_{enc}^{1'} &= LN(MHSA(Z + Pos_{enc}) + Z) \\ f_{enc}^{1} &= LN(FFN(f_{enc}^{1'}) + f_{enc}^{1'}) \\ &\cdots \\ f_{enc}^{l'} &= LN(MHSA(f_{enc}^{l-1}) + f_{enc}^{l-1}) \\ f_{enc}^{l} &= LN(FFN(f_{enc}^{l'}) + f_{enc}^{l'}) \end{aligned} \tag{10}$$

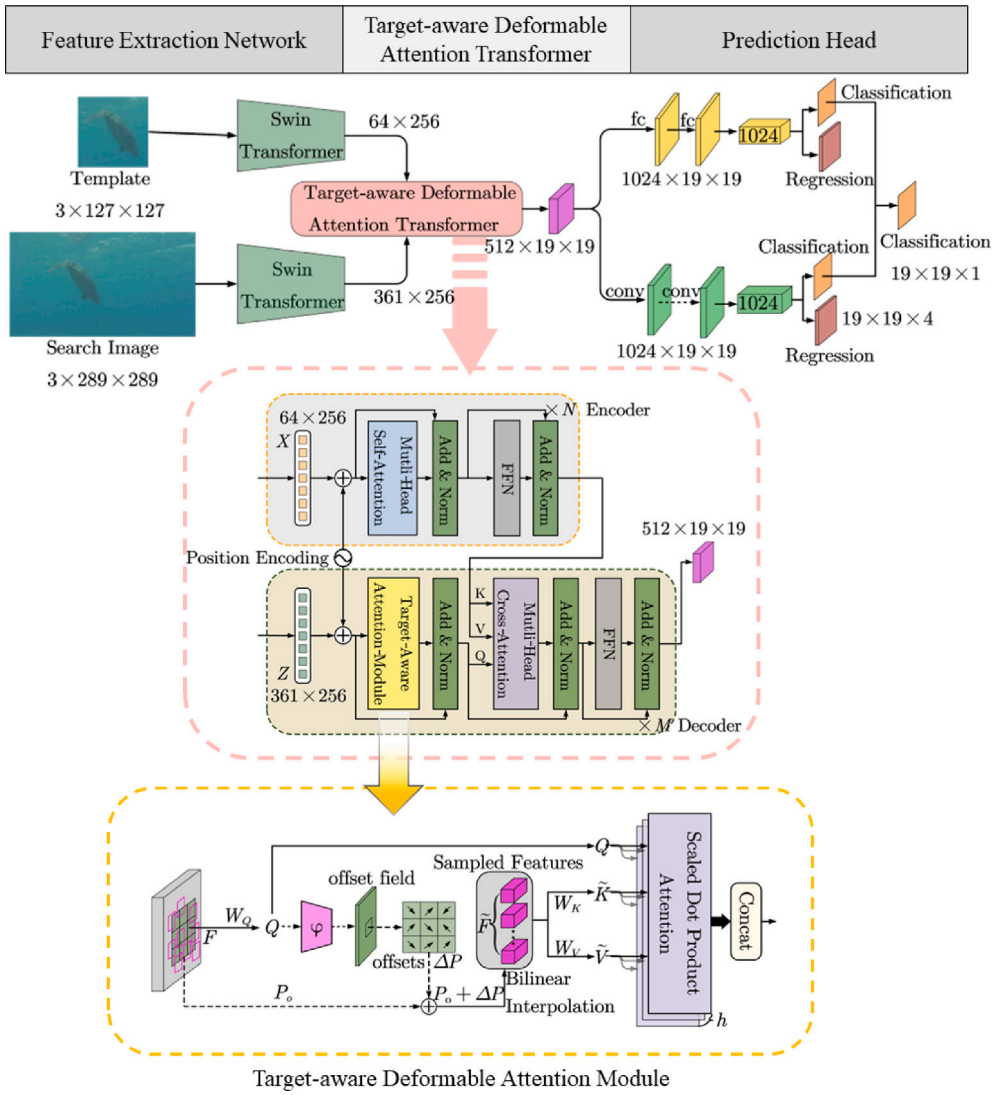


Fig. 4. Proposed framework overview. It consists of three modules: a feature extraction network, target-aware deformable attention transformer, and prediction head. A Swin transformer is used as the backbone network for feature extraction. The target-aware transformer is an encoder–decoder transformer with target-aware deformable attention. The prediction head is a double-head.

where $l \in \{1, \ldots, N\}$ denotes the lth encoder layer, $f_{enc}^{l'} \in \mathbb{R}^{H_z W_z \times C}$ denotes the intermediate output in the encoding process, and f_{enc}^{l-1} denotes the output of the $(l-1)$th encoder layer.

Decoder. The decoder comprises M decoder layers stacked on top of each other. Unlike the decoder in a conventional transformer, in each decoder layer of the proposed approach, the deformable attention of the search region features is first computed by our target-aware deformable attention module. Then, the cross-attention of the search region and template features is computed using native multi-head cross-attention (MHCA). The former is employed to enhance the most relevant target features in the search region, and the latter is utilized to model the correlation between the target and search region features. The process of one decoder layer can be described as follows:

$$
\begin{aligned}
f_{dec}^{1'} &= LN(TAMHSA(X + Pos_{dec}) + X) \\
f_{dec}^{1''} &= LN(MHCA(f_{dec}^{1'}, f_{enc}^{N}) + f_{dec}^{1'}) \\
f_{dec}^{1} &= LN(FFN(f_{dec}^{1''}) + f_{dec}^{1''}) \\
&\cdots \\
f_{dec}^{l'} &= LN(TAMHSA(f_{dec}^{l-1}) + f_{dec}^{l-1}) \\
f_{dec}^{l''} &= LN(MHCA(f_{dec}^{l'}, f_{enc}^{N}) + f_{dec}^{l'}) \\
f_{dec}^{l} &= LN(FFN(f_{dec}^{l''}) + f_{dec}^{l''})
\end{aligned}
\tag{11}
$$

where $X \in \mathbb{R}^{H_x W_x \times C}$ denotes the search region feature, $Pos_{dec} \in \mathbb{R}^{H_x W_x \times C}$ denotes the spatial location encoding, f_{enc}^{N}b denotes the encoded template feature output by the encoder, $l \in \{1, \ldots, M\}$ denotes the lth decoder layer, $f_{dec}^{l'}, f_{dec}^{l''} \in \mathbb{R}^{H_x W_x \times C}$ denotes the intermediate output in the decoding process, and f_{dec}^{l-1} denotes the $(l-1)$th output of the decoder layer.

4.3. Double-head predictor and loss function

The prediction heads of most anchor-based networks can be divided into classification and bounding box regression branches. According to Wu et al. (2020), a fully connected layer is more suitable for a classification branch, whereas a convolutional layer is more appropriate for a regression branch.

Following Wu et al. (2020) and Fu et al. (2022), we adopted the extended version of the double head predictor. The double head includes a fully connected head (fc-head) and convolution head (conv-head). The two heads of the predictor were jointly trained in a supervised manner for both the classification and regression tasks. In the inference, only the conv-head regression score is used to estimate the bounding box, and the classification values from both the fc-head and conv-head

were weighted as the final result. The loss function used in the training was adopted from a previous study (Wu et al., 2020):

$$\mathcal{L}_{total} = w_{fc}\mathcal{L}_{fc} + w_{conv}\mathcal{L}_{conv} \tag{12}$$

where $\mathcal{L}_{fc}$ and $\mathcal{L}_{conv}$ are the fc-head and conv-head losses, respectively. w_{fc} and w_{conv} denote the weights of the fc-head and conv-head, respectively. We set them to 2.0 and 2.5, respectively. Both $\mathcal{L}_{fc}$ and $\mathcal{L}_{conv}$ incorporate the losses for the classification and bounding box regression as follows:

$$\begin{aligned} \mathcal{L}_{fc} &= \lambda\mathcal{L}_{fc}^{cls} + (1-\lambda)\mathcal{L}_{fc}^{reg} \\ \mathcal{L}_{conv} &= (1-\eta)\mathcal{L}_{conv}^{cls} + \eta\mathcal{L}_{cnv}^{reg} \end{aligned} \tag{13}$$

where $\mathcal{L}_{fc}^{cls}$ and $\mathcal{L}_{fc}^{reg}$ denote the classification and bounding box regression losses for the fc-head, $\mathcal{L}_{conv}^{cls}$ and $\mathcal{L}_{conv}^{reg}$ denote the classification and bounding box regression losses for the conv-head, $\mathcal{L}_{conv}^{cls}, \mathcal{L}_{conv}^{reg}$ denote the classification and bounding box regression losses for the conv-head. $\mathcal{L}_{fc}^{cls}, \mathcal{L}_{conv}^{cls}$ employ focal loss, and $\mathcal{L}_{fc}^{reg}, \mathcal{L}_{conv}^{reg}$ use intersection over union (IoU) loss. λ, η are hyperparameter that control the weights of the two losses. In the training, λ was set to 0.7, and η was 0.8. See Sections 5.2.4 and 5.2.5 for specific details on the weights.

We adopted the weighted focal losses $\mathcal{L}_{fc}^{cls}$ and $\mathcal{L}_{conv}^{cls}$ for classification, the Gaussian kernel generates the supervision as follows:

$$G_{xy} = exp(-\frac{\left(x-p_x\right)^2 + \left(y-p_y\right)^2}{2\sigma_p^2}) \tag{14}$$

where (p_x, p_y) denotes the center point coordinates and the standard deviation σ determines the size of the object. The head was then optimized using the focal loss under the supervision of the Gaussian kernel. The focal loss $\mathcal{L}_{fc}^{cls}$ and $\mathcal{L}_{conv}^{cls}$ can be formulated as follows:

$$\mathcal{L}_{fc}^{cls} = \mathcal{L}_{conv}^{cls} = -\sum \begin{cases} (1-\hat{G}_{xy})^{\alpha}\log(\hat{G}_{xy}), & \text{if } G_{xy}=1 \\ (1-G_{xy})^{\beta}(\hat{G}_{xy})^{\alpha}\log(1-\hat{G}_{xy}), & \text{otherwise} \end{cases} \tag{15}$$

Here, $\hat{G}_{xy} \in [0,1]^{W_x \times H_x}$ is the score map, and (W_x, H_x) is the feature size of the search area. α and β are hyperparameters, and we set $\alpha = 2$ and $\beta = 4$ as in Law and Deng (2018).

The loss functions for $\mathcal{L}_{fc}^{reg}$ and $\mathcal{L}_{conv}^{reg}$ are as follows:

$$\mathcal{L}_{fc}^{reg} = \mathcal{L}_{conv}^{reg} = -\sum_{j} ln(IoU(p,g)) \tag{16}$$

where p and g denote the predicted and ground-truth bounding box coordinates, respectively. Given the estimated bounding box p and ground-truth bounding box g, the IoU is defined as $p \cap g / p \cup g$.

5. Experiments

After introducing the implementation details, this section first provides ablation studies to analyze the impact of each component and different design choices. Then, TATrack is compared to other state-of-the-art methods on six different benchmarks.

5.1. Implementation details

This subsection mainly introduces the detailed configurations of training, model, and optimization.

5.1.1. Training datasets

Six training datasets were used in the experiments, including GOT-10K (Huang et al., 2021), LaSOT (Fan et al., 2019), TrackingNet (Muller et al., 2018), COCO (Lin et al., 2014), ILSVRC VID (Russakovsky et al., 2015), and ILSVRC VID (Russakovsky et al., 2015). In addition, when testing on the GOT-10K dataset, we strictly followed the protocol from Huang et al. (2021) and trained only on the GOT-10K (Huang et al., 2021) training dataset. To increase the diversity of the training samples, we randomly scaled the samples using a maximum scale of 0.3. A random translation was performed on the template and search region image pairs in the range of $[-0.2\alpha, 0.2\alpha]$, where $\alpha = \sqrt{(1.3w_z + 0.2h_z) \times (1.3h_z + 0.2w_z)}$ is the template image, and $\alpha = \frac{T \cdot S}{\sqrt{(1.3w_x+0.2h_x)\times(1.3h_x+0.2w_x)}}$ is the search region image. Here, w_z and h_z represent the width and height of the target in the template image, respectively; w_x and h_x denote the width and height of the target in the search region image, respectively; and T, S indicate the size of the template and search region images, respectively. During training, the input sizes of the template and search region were $T = 127$ and $S = 289$, respectively.

5.1.2. Model settings

The proposed method employs the Swin-T (Liu et al., 2021) as the backbone of the feature extraction module. The channels of the hidden layers in the first stage were set to 96, $[2, 2, 6, 2]$ layers are used in each stage, and $[3, 6, 12, 24]$ heads were applied in the MHSA of each stage.

For the MHSA, TAMHSA, and MHCA in the proposed target-aware network, the input feature dimension was set to 256, the number of FFN channels in the hidden layer was set to 2048, the number of heads was eight, and the dropout rate was set to 0.1. The encoder consisted of $N = 2$ layers, and the decoder consisted of $M = 3$ layers. We discuss these hyperparameters in the ablation study.

5.1.3. Optimization

The AdamW optimizer was used in the training process. The training batch size was set to 32, and the learning rate was set to 1e−4 with a decay weight of 1e−4. The model was trained on two NVIDIA 3090 GPUs for a maximum of 20 epochs with 600,000 training image pairs per epoch. The multistep learning rate decay strategy was adopted in the training, and the learning rate was set to decay to 1e−5 and 1e−6 when the number of epochs reached 10 and 15, respectively.

5.2. Ablation study

An ablation study was conducted to analyze the effect of varying the number of layers in the encoder and decoder modules, investigate whether using TAMHSA in the encoder module is necessary, evaluate the effectiveness of TAMHSA, and analyze the weight loss balance. The average overlap (AO) metric estimates the AO between the ground truth and predicted bounding boxes. The $SR_{0.5}$ and $SR_{0.75}$ metrics denote the success rate, which measures the percentage of successfully tracked frames, at which the overlap precisions exceed thresholds of 0.50 and 0.75, respectively.

5.2.1. Number of encoder and decoder layers

Table 1 lists the performance of the proposed method with respect to the different layers of the encoder and decoder on GOT-10K. As shown in Table 1, when the number of encoder layers is fixed, an increase in the number of decoder layers ($M > 3$) leads to performance degradation. In addition, when the number of decoder layers is fixed, an increase in the number of encoder layers ($N > 2$) leads to performance degradation. Therefore, the number of encoder layers was set to two, and the number of decoder layers was set to three in the remaining experiments.

5.2.2. TAMHSA in the encoder

This subsection explores whether TAMHSA is required for both the encoder and decoder. As shown in Table 2, the performance of the encoder with TAMSA degrades significantly compared with that of traditional MHSA. This is because of the low resolution of the features in the template branch and information loss caused by TAMHSA. Therefore, an encoder using the traditional transformer and a decoder with TAMHSA is the best architectural setting.

Table 1
Performances of the encoder and decoder with different numbers of layers tested on GOT-10K.

Encoder-Decoder layer	AO	$SR_{0.5}$	$SR_{0.75}$	Speed (fps)
(2, 2)	0.675	0.771	0.622	30
(2, 3)	0.697	0.793	0.642	28
(2, 4)	0.670	0.762	0.619	25
(1, 3)	0.673	0.767	0.620	30
(3, 3)	0.684	0.772	0.631	24
(4, 3)	0.674	0.769	0.624	23

Table 2
Performances of the encoder and decoder with different numbers of layers with and without TAMHSA tested on GOT-10K.

Encoder	Decoder	Layers	AO	$SR_{0.5}$	$SR_{0.75}$	Speed (fps)
		(2, 3)	0.671	0.763	0.616	32
	✓	(2, 3)	0.697	0.793	0.642	30
✓	✓	(1, 2)	0.665	0.754	0.603	35
✓	✓	(2, 2)	0.678	0.773	0.619	27
✓	✓	(2, 3)	0.674	0.769	0.634	24
✓	✓	(3, 3)	0.674	0.769	0.620	25

Table 3
Performance comparison of different methods on GOT-10K. SA denotes traditional self-attention; TS-SA, top-k sparse self-attention; and PS-SA, probability sparse self-attention.

SA	TS-SA	PS-SA	Def-SA	AO	$SR_{0.5}$	$SR_{0.75}$	Speed (fps)
√				0.671	0.763	0.616	32
	√			0.649	0.741	0.583	28
		√		0.683	0.786	0.630	32
			√	0.697	0.793	0.642	28

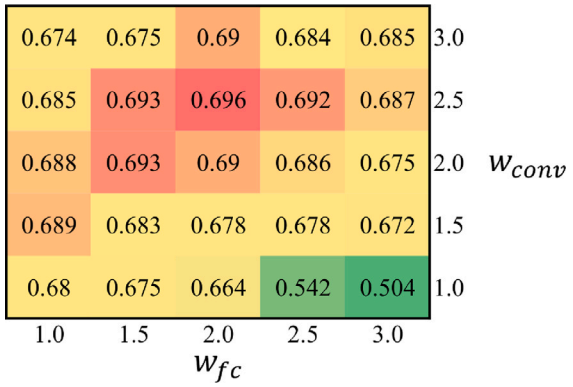


Fig. 5. AO for different choices of w_{fc} and w_{conv} on GOT-10K.

5.2.3. TAMHSA effectiveness

To validate the effectiveness of deformable attention in TAMHSA. We compared the performance of the proposed method to that using standard self-attention (Vaswani et al., 2017), top-k sparse self-attention (Fu et al., 2022), and probability sparse self-attention (Zhou et al., 2021) approaches. All methods were trained and tested on the GOT-10K training and testing sets. As demonstrated in Table 3, the proposed approach achieved the highest performance, empirically proving the effectiveness of the proposed TAMHSA.

5.2.4. Balance weights w_{fc} and w_{conv}

Fig. 5 illustrates the performance of AO for different choices of w_{fc} and w_{conv} in the GOT-10K dataset. Here, w_{fc} represents the weights of the fc-head, which were used to the classification task to train the network's discriminative ability, while w_{conv} represents the weights of the conv-head, which were used to the regression task to train the network's target localization ability. The goal of object tracking is to continuously track and locate the target in subsequent frames, given an initial frame. Therefore, the weight of the regression loss was larger than that of the classification loss. In order to achieve the best training results by balancing the losses effectively (Fig. 5), we tested models trained with different weights w_{fc} and w_{conv} on the GOT-10K dataset. As shown in Fig. 5, when w_{fc} is greater than w_{conv}, AO decreases as the weight of w_{fc} increases. This is because of the low weight of w_{conv}, which leads to an imbalance in the losses as w_{fc} increases, resulting in lower localization accuracy of the trained model. When w_{fc} is less than w_{conv}, a weight of 2.5 for w_{conv} yields the best results. The ablation experiment results demonstrate that the weights $w_{fc} = 2.0$ and $w_{conv} = 2.5$ yield the optimal performance.

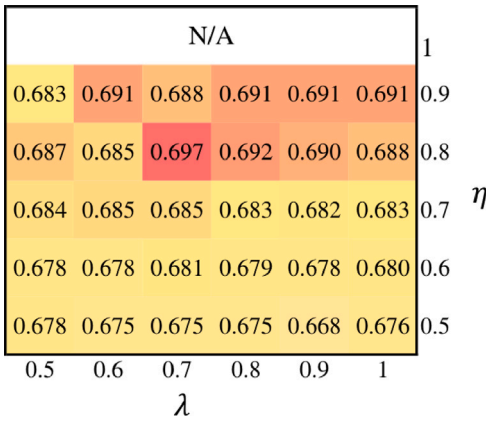


Fig. 6. AO for different choices of λ and η on GOT-10K. We trained a double-head model for each (λ, η) pair.

5.2.5. Balance weights λ and η

The λ parameter is used to adjust the weight of the fully connected head losses and the η parameter is used to adjust the weight of the convolutional head losses. Fig. 6 shows the test results for different weights of λ and η on the GOT-10K dataset. The weight combination of $\lambda = 0.7$ and $\eta = 0.8$ was determined to be optimal.

5.3. Comparisons with state-of-the-art trackers

To demonstrate the effectiveness of the proposed models, we compared them with state-of-the-art trackers using six benchmarks: GOT-10K, TrackingNet, LaSOT, OTB100, UAV123, and NFS30. As shown in Table 4, we observed the following regarding the datasets.

GOT-10K: One of the challenges addressed by the dataset is long-term tracking. It focuses on tracking objects across numerous frames. This dataset provides high-quality annotations with per-frame bounding boxes for a diverse set of object classes.

TrackingNet: This dataset aims to address the limitations of previous datasets by addressing issues such as target initialization and long-term tracking. It consists of a large-scale dataset with more than 30 object classes and per-object bounding box annotations, making it suitable for various tracking scenarios.

LaSOT: This dataset focuses on addressing the challenges of large-scale object tracking, including variations in the scale, viewpoint, and occlusion. It contains a comprehensive list of 1400 videos with a diverse set of objects, annotations for tight and full bounding boxes, and extensive evaluation metrics.

OTB100: This is a well-known benchmark dataset designed to evaluate the performance of tracking algorithms. It covers various aspects of object tracking and consists of 100 fully annotated video sequences with various challenges such as illumination changes, occlusions, and motion variations.

Table 4
Comparison of current datasets for object tracking. # denotes the number of corresponding items, and FR denotes the frame rate.

Datasets	#Videos	#Min	#Mean	#Max	#Total	#FR	#Attributes
OTB100	100	71	590	3872	59K	30 fps	11
UAV123	123	109	915	3085	113K	30 fps	12
NFS	100	169	3830	20 665	383K	240 fps	9
TrackingNet	30,643	–	480	–	14.43M	30 fps	15
GOT-10K	10,000	29	149	1418	1.5M	10 fps	6
LaSOT	1400	1000	2506	11 397	3.52M	30 fps	14

Table 5
Performances of our method and existing state-of-the-art methods on the **GOT-10k** test set in terms of the average overlap(AO) and success rates($SR_{0.5}$ and $SR_{0.75}$) at threshold 0.5 and 0.75. Red and blue numbers indicate the best and second-best results, respectively.

Trackers	Source	AO	$SR_{0.5}$	$SR_{0.75}$
TATrack	Ours	0.697	0.793	0.642
CSWinTT (Song et al., 2022)	CVPR2022	0.694	0.789	0.654
SparseTT (Fu et al., 2022)	CVPR2022	0.693	0.791	0.638
STARK-ST101 (Yan et al., 2021)	ICCV2021	0.688	0.781	0.641
TransT (Chen et al., 2021)	CVPR2021	0.671	0.768	0.609
TrDiMP (Wang et al., 2021)	CVPR2021	0.671	0.777	0.583
AutoMatch (Zhang et al., 2021)	ECCV2021	0.652	0.766	0.543
STMTrack (Fu et al., 2021)	CVPR2021	0.642	0.737	0.575
KYS (Bhat et al., 2020)	ECCV2020	0.636	0.751	0.515
DTT (Yu et al., 2021)	ICCV2021	0.634	0.749	0.514
PrDiMP (Danelljan et al., 2020)	CVPR2020	0.634	0.738	0.543
DiMP50 (Bhat et al., 2019)	ICCV2019	0.611	0.717	0.492
SiamFC++ (Xu et al., 2020)	AAAI2020	0.595	0.695	0.479

Table 6
Performances of our approach and existing schemes on **TrackingNet**, where "AUC", "PR", and "NPR" indicate the area under the curve, precision, and normalized precision, respectively. Red and blue numbers indicate the best and second-best results, respectively.

Trackers	Source	AUC (%)	PR (%)	NPR (%)
TATrack	Ours	82.1	80.1	86.9
CSWinTT (Song et al., 2022)	CVPR2022	81.9	79.5	86.7
SparseTT (Fu et al., 2022)	CVPR2022	81.7	79.5	86.6
STARK-ST50 (Yan et al., 2021)	ICCV2021	81.3	–	86.1
TransT (Chen et al., 2021)	CVPR2021	81.4	80.3	86.7
STMTrack (Fu et al., 2021)	CVPR2021	80.3	76.7	85.1
DTT (Yu et al., 2021)	ICCV2021	79.6	78.9	85.0
TrDiMP (Wang et al., 2021)	CVPR2021	78.4	73.1	83.3
AutoMatch (Zhang et al., 2021)	ECCV2021	76.0	72.5	82.4
PrDiMP (Danelljan et al., 2020)	CVPR2020	75.8	70.4	81.6
SiamFC++ (Xu et al., 2020)	AAAI2020	75.4	70.5	80.0
KYS (Bhat et al., 2020)	ECCV2020	74.0	68.8	80.0
DiMP50 (Bhat et al., 2019)	ICCV2019	74.0	68.7	80.1

UAV123: This dataset targets challenges related to tracking objects from the perspective of unmanned aerial vehicles (UAVs) and drones. The dataset includes 123 video sequences captured by UAVs and covers difficulties such as rapid motion, scale variations, and object occlusions.

NFS: This dataset was designed to evaluate tracking algorithms at high operating speeds. The dataset provides 1000 high frame rate video sequences at a frame rate of 240 fps, which can be used to accurately analyze the effect of changes in the target appearance on the tracking algorithm.

5.3.1. GOT-10K

To guarantee a fair comparison, we strictly followed the protocol of GOT-10K, and all compared trackers were trained and tested only on the training and test sets of GOT-10K. Twelve recent trackers were included in the comparison. Most of these trackers are transformer-based, including CSWinTT (Song et al., 2022), SparseTT (Fu et al., 2022), STARK-ST101 (Yan et al., 2021), TransT (Chen et al., 2021), TrDiMP (Wang et al., 2021), and DTT (Yu et al., 2021). As shown in Table 5, our method outperformed all existing methods. In particular, the AO and SR0.5 achieved by TATrack were 69.7% and 79.3%, respectively, which were 0.3% and 0.4% higher than those of the second-best method, CSWinTT. Compared with TransT, the performance of the proposed method was 2.6%, 2.5%, and 3.3% higher in terms of the AO, $SR_{0.5}$, and $SR_{0.75}$, respectively. These results demonstrate the strong generality of the proposed approach.

5.3.2. TrackingNet

TrackingNet (Muller et al., 2018) is a large-scale dataset used for target tracking in the wild. It contains over 30,000 video sequences and 14 million dense bounding-box annotations. The test performance results of all trackers were verified on an online verification server. As shown in Table 6, our method outperformed all comparison trackers. The area under the curve (AUC), precision, and normalized precision scores achieved using the proposed approach were 82.1%, 80.1%, and 86.9%, respectively. These results demonstrate that the proposed method can achieve a state-of-the-art performance in large-scale tracking scenarios.

5.3.3. LaSOT

LaSOT is a densely annotated large-scale dataset containing 1400 long-term video sequences and 14 attributes, such as full occlusion, partial occlusion, deformation, and motion blur. As shown in Table 7, our method achieved an AUC score of 66.1% and accuracy

Table 7
Performances of our approach and existing schemes on **LaSOT**, where "AUC" "PR" and "NPR" indicate the success, precision, and normalized precision, respectively. Red and blue numbers indicate the best and second-best results, respectively.

Trackers	Source	AUC (%)	PR (%)	NPR (%)
TATrack	Ours	66.1	70.3	69.4
SparseTT (Fu et al., 2022)	CVPR2022	66.0	70.1	74.8
TransT (Chen et al., 2021)	CVPR2021	64.9	69.0	73.8
TrDiMP (Wang et al., 2021)	CVPR2021	63.9	61.4	–
TrSiam (Wang et al., 2021)	CVPR2021	62.4	60.0	–
STMTrack (Fu et al., 2021)	CVPR2021	60.6	63.3	69.3
DTT (Yu et al., 2021)	ICCV2021	60.1	–	–
PrDiMP (Danelljan et al., 2020)	CVPR2020	59.8	60.9	–
AutoMatch (Zhang et al., 2021)	ECCV2021	58.3	59.9	67.5
DiMP50 (Bhat et al., 2019)	ICCV2019	56.9	56.7	65.0
KYS (Bhat et al., 2020)	ECCV2020	55.4	–	63.3
SiamFC++ (Xu et al., 2020)	AAAI2020	54.3	54.7	62.3
SiamRPN++ (Li et al., 2019)	CVPR2019	49.6	49.1	56.9

Table 8
Comparison of the AUC scores of state-of-the-art trackers on commonly used small-scale datasets, including **UAV123**, **NFS30**, **OTB100**. red and blue numbers indicate the best and second-best results, respectively.

Trackers	Source	Datasets			FPS
		OTB100	UVA123	NFS30	
TATrack	Ours	71.8	69.4	65.1	30
SparseTT (Fu et al., 2022)	CVPR2022	70.4	70.4	–	40
STARK-ST50 (Yan et al., 2021)	ICCV2021	68.5	69.1	65.2	42
TransT (Chen et al., 2021)	CVPR2021	69.4	69.1	65.7	50
PrDiMP50 (Danelljan et al., 2020)	CVPR2021	69.6	68.0	63.5	30
SiamRPN++ (Li et al., 2019)	CVPR2019	69.6	61.3	50.2	35

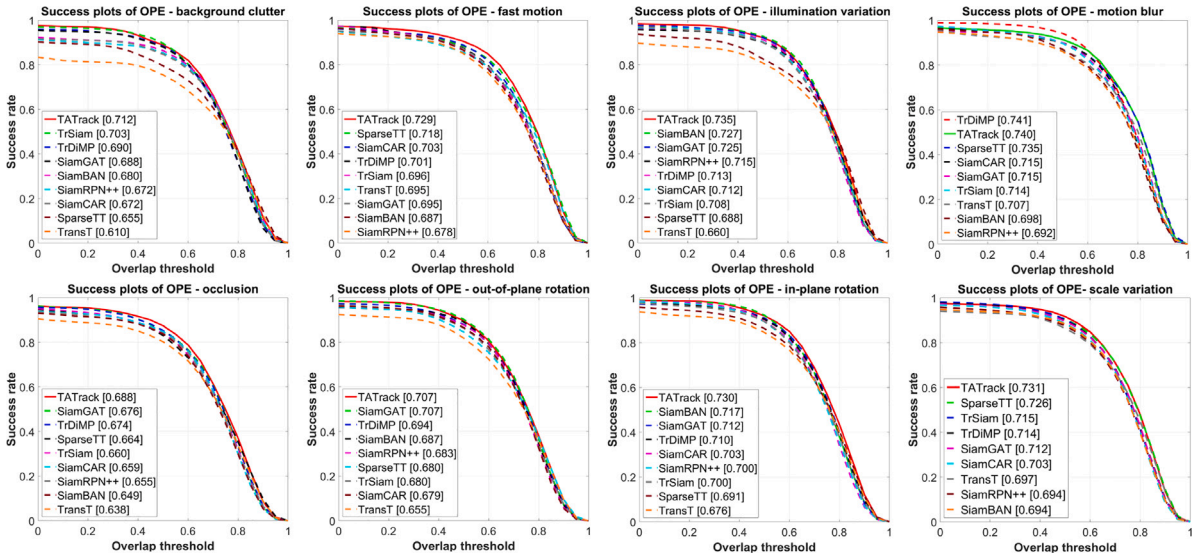


Fig. 7. Comparison of AUC scores for the 8 attributes in OTB100, including BC, FM, IV, MB, OCC, OPR, IPR, and SV.

of 70.3%, outperforming the state-of-the-art methods. The normalized precision scores were comparable. These results demonstrate the significant potential of the proposed method for long-term tracking.

5.3.4. OTB100

OTB100 contains 100 short-term video sequences with 11 tracking attributes. The performances of the existing trackers on this benchmark have become saturated in recent years. However, as listed in Table 8, we can outperform the state-of-the-art methods by a large margin. These results demonstrate the excellent performance of the proposed approach for small-scale, short-term object tracking.

5.3.5. UAV123

The UAV123 dataset contains 123 video sequences and more than 110,000 frames. These were captured using a camera mounted on a drone and includes all bounding boxes and property annotations. As shown in Table 8, our method achieved the second-best performance with an AUC score of 69.4%. The results illustrate that the proposed approach is superior to most existing studies on UAV object-tracking tasks.

5.3.6. NFS30

Need for Speed (NFS) is a dataset containing 100 videos of fast-moving objects at 30 and 120 fps. We tested on a test dataset of 30 frames (NFS30). The results shown in Table 8 show that our method can achieve a performance comparable to that of existing methods.

5.4. Efficiency comparison

Table 9 compares the proposed method with existing transformer-based trackers in terms of speed and number of parameters. The

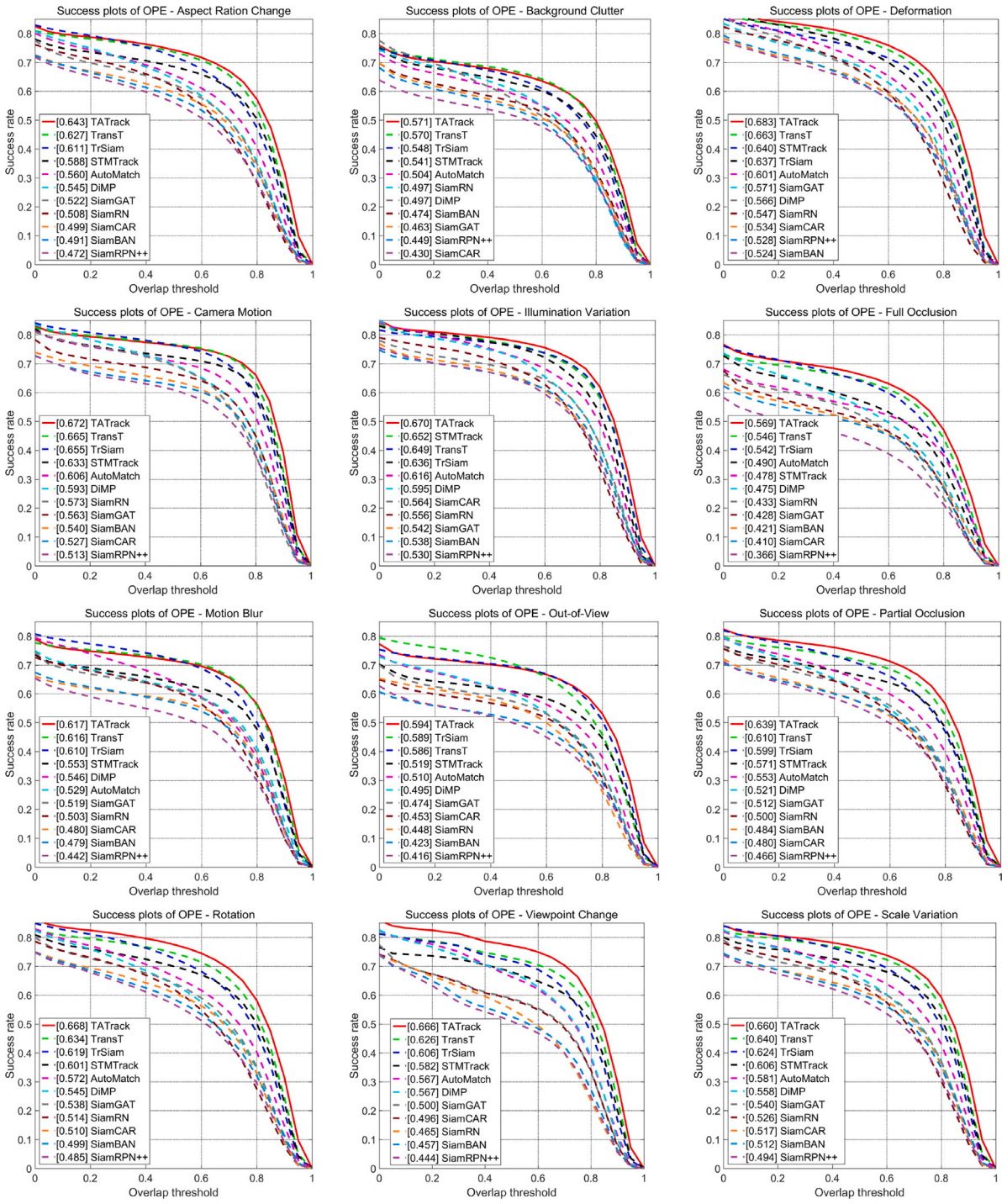


Fig. 8. Comparison of AUC scores for the 12 attributes in LaSOT, including ARC, BC, DEF, CM, IV, FOC, MB, OV, POC, ROT, VC, and SV.

Table 9
Comparison of different transformer-based trackers regarding the running speed and number of parameters. MACs denotes multiply-accumulate computations, and Params denotes the number of parameters.

Trackers	Speed (fps)	MACs (G)	Params (M)
TransT (Chen et al., 2021)	50	–	23
SparseTT (Fu et al., 2022)	40	10.9	28
STARK-ST50 (Yan et al., 2021)	42	10.9	24
STARK-ST101 (Yan et al., 2021)	32	18.5	42
Only transformer	32	10.9	28
Ours	30	10.7	27

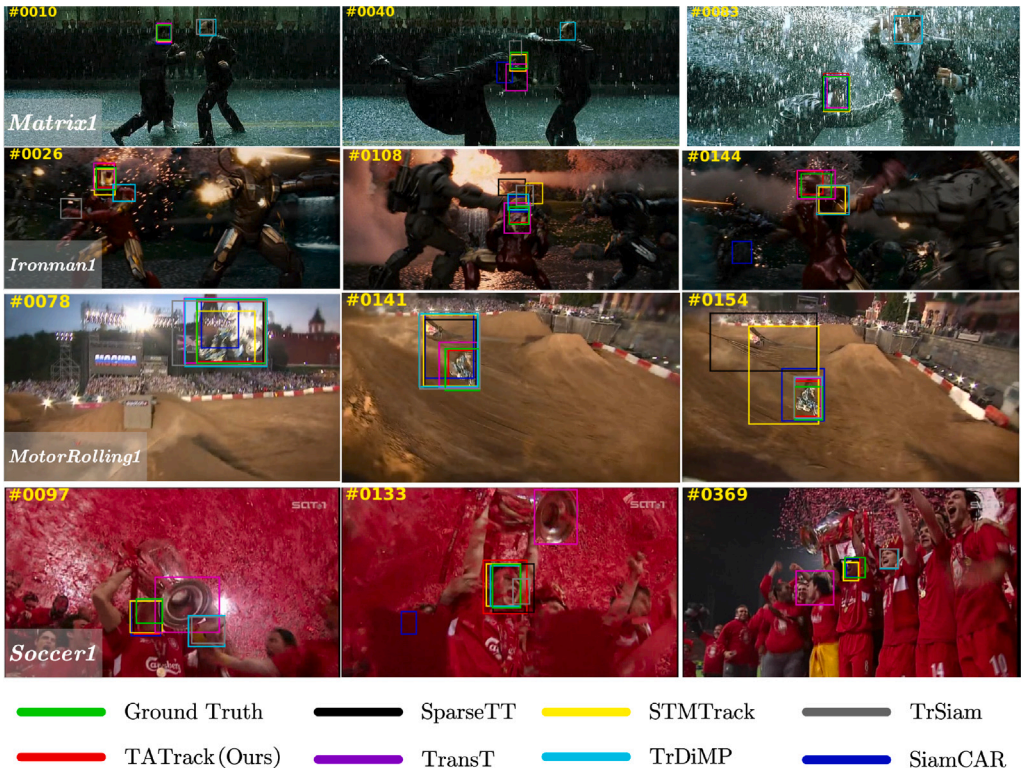


Fig. 9. Visualization of tracking results. We visualize the tracking results for three frames of four video sequences using different trackers, including *Matrix1*, *Ironman1*, *MotorRolling1*, and *Soccer1*.

table displays the number of parameters (Params), multiply-accumulate computations (MACs), and the frame rate. The proposed method was run at a real-time speed of 30 FPS. Compared with STARK-ST101, the proposed method ran at a comparable speed but with 15 million fewer parameters and 3.8 G less of computation overhead. Moreover, as shown in Table 5, our performance was far better than that of STARK-ST101.

5.5. Attribution based evaluation

To further validate the performance of our approach in different scenarios, we tested our method on different tracking attributes of the LaSOT and OTB100 datasets. Using the LaSOT dataset, we evaluated the performance of 12 attributes: aspect ratio change (ARC), background clutter (BC), camera motion (CM), deformation (DEF), full occlusion (FOC), illumination variation (IV), motion blur (MB), out-of-view (OV), partial occlusion (POC), rotation (ROT), scale variation (SV), and viewpoint change (VC). As shown in Fig. 8, TATrack achieves the best performance among all 12 attributes of LaSOT. Similarly, we evaluated the performance of eight attributes from the OTB100 dataset: BC, fast motion (FM), IV, MB, occlusion (OCC), out-of-plane rotation (OPR), in-plane rotation (IPR), and SV. As illustrated in Fig. 7, compared with eight trackers, SparseTT (Fu et al., 2022), TransT (Chen et al., 2021), TrSiam (Wang et al., 2021), TrDiMP (Wang et al., 2021), SiamGAT (Guo et al., 2021), SiamBAN (Chen et al., 2020), SiamRPN++ (Li et al., 2019), and SiamCAR (Guo et al., 2020), the proposed method outperforms these methods in seven tracking attributes. For MB, the TATrack was only 0.1% lower than TrDiMP. The TATrack achieved 1.2% and 1.3% higher values than the second best for the OCC and IPR attributes, respectively. Because of target-aware attention, the TATrack focuses more on target information and suppresses background distractions, thereby enabling our method to effectively address large target appearance variations (see Fig. 8).

5.6. Qualitative analysis

As shown in Fig. 9, we qualitatively compared the proposed method on four challenging video sequences with other existing state-of-the-art trackers, including TransT (Chen et al., 2021), SparseTT (Fu et al., 2022), TrSiam (Wang et al., 2021), TrDiMP (Wang et al., 2021), STMTrack (Fu et al., 2021), and SiamCAR (Guo et al., 2020). The four video sequences were *Matrix1*, *Ironman1*, *MotorRolling1*, and *Soccer1*. They have many challenging attributes, such as BC, CM, DEF, FM, FOC, and IV.

As shown in Fig. 9, *Matrix1* faces obstacles such as similar object interference and background distractions. With similarity object interference at frame #10, TrDiMP and TrSiam exhibited tracking errors, whereas our approach tracked accurately. In addition, most existing trackers fail to track when the target appearance changes and motion blur occurs at frame #40; our method tracked this successfully and accurately. With the large illumination change in frame #83, the proposed tracker worked well.

Similar to the challenging properties of *Matrix1*, *Ironman1* suffers from a large amount of background clutter, deformation, motion blur, and illumination changes; however, our tracker still tracked successfully. In various camera-movement scenarios, the proposed tracker accurately estimated the target location. In *MotorRolling1*, fast motion and rotation are significant challenges. However, our approach tracked the target well and consistently, whereas other trackers fail or generate bounding boxes with low accuracies.

With numerous partially and fully occluded cases in *Soccer1*, the TATrack can accurately estimate the target. In particular, the performance of our tracker was particularly impressive for frames #97 and #369. All these results demonstrate the excellent tracking performance of the TATrack in various scenarios.

6. Conclusion

Although the transformer has achieved state-of-the-art performances for various vision tasks owing to the strong representation power achieved by its self-attention mechanism, directly using self-attention for visual object tracking typically is suboptimal, as self-attention pays equal attention to both the foreground and background.

This study integrated deformable attention into the tracker to allow the model to focus more on target-related information and suppress background distractions in the search region. In addition to deformable attention, a transformer-based Siamese tracking architecture, TATrack, was proposed. The TATrack consists of a feature extraction module, target-aware transformer, and prediction head.

Extensive experiments are conducted using public benchmarks. The results demonstrate that the proposed tracker can achieve state-of-the-art performances on large-scale datasets and comparable performances on small-scale datasets.

CRediT authorship contribution statement

Kai Huang: Conceptualization, Methodology, Software, Investigation, Data curation, Writing – original draft, Visualization. **Jun Chu:** Validation, Project administration, Funding acquisition. **Lu Leng:** Resources, Supervision, Project administration, Funding acquisition. **Xingbo Dong:** Formal analysis, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jun Chu reports financial support was provided by National Natural Science Foundation of China (No.62162045). Jun Chu reports financial support was provided by Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai (No.AI2020004). Lu Leng reports financial support was provided by Technology Innovation Guidance Program Project (Special Project of Technology Cooperation, Science and Technology Department of Jiangxi Province, No.20212BDH81003). Lu Leng reports financial support was provided by National Natural Science Foundation of China (No.61866028).

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62162045), the National Natural Science Foundation of China (No. 61866028), Technology Innovation Guidance Program Project, China (Special Project of Technology Cooperation, Science and Technology Department of Jiangxi Province, No. 20212BDH81003). Jun Chu reports financial support was provided by Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai (No. AI2020004).

References

Bagherzadeh, S.Z., Toosizadeh, S., 2022. Eye tracking algorithm based on multi model Kalman filter. HighTech Innov. J. 3 (1), 15–27.

Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S., 2016. Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision. Springer, pp. 850–865.

Bhat, G., Danelljan, M., Gool, L.V., Timofte, R., 2019. Learning discriminative model prediction for tracking. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6181–6190.

Bhat, G., Danelljan, M., Gool, L.V., Timofte, R., 2020. Know your surroundings: Exploiting scene information for object tracking. In: European Conference on Computer Vision.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: European Conference on Computer Vision (ECCV). Springer, pp. 213–229.

Chen, B., Li, P., Bai, L., Qiao, L., Shen, Q., Li, B., Gan, W., Wu, W., Ouyang, W., 2022. Backbone is all your need: A simplified architecture for visual object tracking. In: European Conference on Computer Vision (ECCV). Springer, pp. 375–392.

Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H., 2021. Transformer tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8126–8135.

Chen, Z., Zhong, B., Li, G., Zhang, S., Ji, R., 2020. Siamese box adaptive network for visual tracking. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6667–6676.

Cui, Y., Cheng, J., Wang, L., Wu, G., 2022. MixFormer: End-to-end tracking with iterative mixed attention. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13598–13608.

Danelljan, M., Gool, L.V., Timofte, R., 2020. Probabilistic regression for visual tracking. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7181–7190.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR.

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H., 2019. Lasot: A high-quality benchmark for large-scale single object tracking. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5369–5378.

Fu, Z., Fu, Z., Liu, Q., Cai, W., Wang, Y., 2022. Sparsett: Visual tracking with sparse transformers. In: International Joint Conferences on Artificial Intelligence.

Fu, Z., Liu, Q., Fu, Z., Wang, Y., 2021. Stmtrack: Template-free visual tracking with space-time memory networks. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13769–13778.

Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., Shen, C., 2021. Graph attention tracking. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9538–9547.

Guo, D., Wang, J., Cui, Y., Wang, Z., Chen, S., 2020. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6268–6276.

Guo, M., Zhang, Z., Fan, H., Jing, L., Lyu, Y., Li, B., Hu, W., 2022. Learning target-aware representation for visual tracking via informative interactions. In: International Joint Conference on Artificial Intelligence (IJCAI).

Huang, L., Zhao, X., Huang, K., 2021. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Trans. Pattern Anal. Mach. Intell. 43, 1562–1577.

Javed, S., Danelljan, M., Khan, F.S., Khan, M.H., Felsberg, M., Matas, J., 2022a. Visual object tracking with discriminative filters and siamese networks: A survey and outlook. IEEE Trans. Pattern Anal. Mach. Intell. 1–20.

Javed, S., Danelljan, M., Khan, F.S., Khan, M.H., et al., 2022b. Visual object tracking with discriminative filters and siamese networks: A survey and outlook. IEEE Trans. Pattern Anal. Mach. Intell. 1–20. http://dx.doi.org/10.1109/TPAMI.2022.3212594.

Jiao, L., Wang, D., Bai, Y., Chen, P., Liu, F., 2021. Deep learning in visual tracking: A review.. IEEE Trans. Neural Netw. Learn. Syst. 1–20.

Khan, S.H., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M., 2022. Transformers in vision: A survey. ACM Comput. Surv. 54, 1–41.

Kurdthongmee, W., Kurdthongmee, P., Suwannarat, K., Kiplagat, J.K., 2022. A YOLO detector providing fast and accurate pupil center estimation using regions surrounding a pupil. Emerg. Sci. J. 6 (5), 985–997.

Law, H., Deng, J., 2018. Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 734–750.

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J., 2019. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4277–4286.

Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X., 2018. High performance visual tracking with siamese region proposal network. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8971–8980.

Lin, L., Fan, H., Xu, Y., Ling, H., 2021. Swintrack: A simple and strong baseline for transformer tracking. arXiv preprint arXiv:2112.00995.

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

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9992–10002.

Ma'arif, A., Raharja, N.M., Supangkat, G., Arofiati, F., Sekhar, R., Rijalusalam, D.U., et al., 2021. Pid-based with odometry for trajectory tracking control on four-wheel omnidirectional covid-19 aromatherapy robot. Emerg. Sci. J. 5, 157–181.

Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B., 2018. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 300–317.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. 115, 211–252.

Song, Z., Yu, J., Chen, Y.-P.P., Yang, W., 2022. Transformer tracking with cyclic shifting window attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8791–8800.

Tian, Z., Shen, C., Chen, H., He, T., 2019. FCOS: Fully convolutional one-stage object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9626–9635.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, Vol. 30.

Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.S., 2019. Fast online object tracking and segmentation: A unifying approach. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1328–1338.

Wang, N., gang Zhou, W., Wang, J., Li, H., 2021. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1571–1580.

Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y.R., 2020. Rethinking classification and localization for object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10183–10192.

Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G., 2022. Vision transformer with deformable attention. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4784–4793.

Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G., 2020. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. pp. 12549–12556.

Yan, B., Peng, H., Fu, J., Wang, D., Lu, H., 2021. Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10448–10457.

Yang, K., Zhang, H., Gao, F., Shi, J., Zhang, Y., Wu, Q.J., 2022. DETA: A point-based tracker with deformable transformer and task-aligned learning. IEEE Trans. Multimed. 1–14.

Ye, B., Chang, H., Ma, B., Shan, S., Chen, X., 2022. Joint feature learning and relation modeling for tracking: A one-stream framework. In: European Conference on Computer Vision (ECCV). Springer, pp. 341–357.

Yu, B., Tang, M., Zheng, L., Zhu, G., Wang, J., Feng, H., Feng, X., Lu, H., 2021. High-performance discriminative tracking with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9856–9865.

Zhang, Z., Liu, Y., Wang, X., Li, B., Hu, W., 2021. Learn to match: Automatic matching network design for visual tracking. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13319–13328.

Zhang, Z., Peng, H., Fu, J., Li, B., Hu, W., 2020. Ocean: Object-aware anchor-free tracking. In: European Conference on Computer Vision (ECCV). Springer, pp. 771–787.

Zhang, Z., Peng, H., Wang, Q., 2019. Deeper and wider siamese networks for real-time visual tracking. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4586–4595.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W., 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. pp. 11106–11115.