electronics

MDPI

Article

Consistent Weighted Correlation-Based Attention for Transformer Tracking

Lei Liu [1], Genwen Fang [1], Jun Wang [2], Shuai Wang [1,*], Chun Wang [1], Longfeng Shen [1,3], Kongfen Zhu [1] and Silas N. Melo [4]

[1] School of Computer Science and Technology, Anhui Engineering Research Center for Intelligent Computing and Application on Cognitive Behavior (ICACB), Huaibei Normal University, Huaibei 235000, China; liul@chnu.edu.cn (L.L.); 12211080780@chnu.edu.cn (G.F.); chunwang1988@chnu.edu.cn (C.W.); shenlf5007@chnu.edu.cn (L.S.); zhukf@chnu.edu.cn (K.Z.)
[2] College of Electronic and Information Engineering, Hebei University, Baoding 071000, China; junwanghbu@hbu.edu.cn
[3] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China
[4] Department of Geography, Universidade Estadual do Maranhão, São Luís 65055-000, Brazil; silasmelo@professor.uema.br
* Correspondence: wangshuai@chnu.edu.cn

Abstract: Attention mechanism takes a crucial role among the key technologies in transformer-based visual tracking. However, the current methods for attention computing neglect the correlation between the query and the key, which results in erroneous correlations. To address this issue, a CWCTrack framework is proposed in this study for transformer visual tracking. To balance the weights of the attention module and enhance the feature extraction of the search region and template region, a consistent weighted correlation (CWC) module is introduced into the cross-attention block. The CWC module computes the correlation score between each query and all keys. Then, the correlation multiplies the consistent weights of the other query–key pairs to acquire the final attention weights. The weights of consistency are computed by the relevance of the query–key pairs. The correlation is enhanced for the relevant query–key pair and suppressed for the irrelevant query–key pair. Experimental results conducted on four prevalent benchmarks demonstrate that the proposed CWCTrack yields preferable performances.

Keywords: consistent weighted correlation; vision transformer; attention; transformer tracking

Citation: Liu, L.; Fang, G.; Wang, J.; Wang, S.; Wang, C.; Shen, L.; Zhu, K.; Melo, S.N. Consistent Weighted Correlation-Based Attention for Transformer Tracking. *Electronics* **2023**, *12*, 4648. https://doi.org/10.3390/electronics12224648

Academic Editors: Haibin Wu, Aili Wang and Yuji Iwahori

Received: 29 September 2023
Revised: 7 November 2023
Accepted: 13 November 2023
Published: 15 November 2023

1. Introduction

Object tracking techniques have witnessed extensive interest and research in computer vision in recent years [1,2]. Given a certain target object in the initial video frame, the tracking algorithms first extract the features of the target and analyze the region of interest; then, similar features among the regions of interest are sought in the following frames; and finally, the tracker pursues the location of the target in the subsequent frames.

In conventional object tracking methods, early Siamese-based trackers [3–6] first employ two convolutional neural network (CNN) backbones with shared structures and parameters to retrieve the features of the template and the search regions. Then, the correlation-based network is adopted to calculate the similarity between the template and the search regions. However, these CNN-based feature extractions usually solely focus on local areas, lacking a global understanding of the surroundings of the target object. This may lead to failure in tracking complex scenarios, such as target occlusion, deformation, or scaling [5].

Therefore, recent mainstream tracking methods [7–11] have introduced transformers [12] for target tracking. Among them, TransT [7] adopts a framework similar to the Siamese-based tracker but uses a transformer for feature fusion, thereby achieving sufficient interaction of the target information. A reconstruction patch strategy is proposed

in [8], which combines the extracted features with multiple spatial dimension elements to form a new patch, replacing the feature fusion layer in TransT. MixFormer [9] proposed the mixture attention module (MAM), which allows for the simultaneous extraction of target-specific features and extensive communication between the target and the search region. OSTrack [10] connects flat templates with search regions and feeds them back into a series of self-attention layers for joint feature learning and relationship modeling. A deformable transformer tracking (D-TransT) is proposed in [11], which uses a deformable attention module that pre-filters all the prominent key elements in the feature map using a small set of sampling positions. The module can naturally expand to aggregate multi-scale features.

The attention mechanism introduces a self-attention process, which facilitates the model to dynamically explore the correlation between various positions in the image sequences and focus more on the key regions for the tracking task. There are two kinds of attention: self-attention enforces the feature representation of the template and search region, and cross-attention establishes dependencies between the template and search region for object prediction.

However, the conventional transformer computes the correlation between each query–key pair independently via the dot product while ignoring the correlation between other query–key pairs. This may lead to inaccurate correlation calculations. This imprecise correlation may further lead the attention mechanism to excessively focus on the background or ignore the important target. For example, if the attention mechanism mistakenly associates a key of an interfering object or background region when paying attention to the target position, the tracker may produce incorrect results.

To deal with these aforementioned challenges, we propose a consistent weighted correlation (CWC) module to promote the feature representation ability of the template and search region. Due to the consistency between the query and its correspondence key, the correlations between relevant query–key pairs should coincide with each other. For example, a key has a high correlation with a query, and its adjacent keys will also have a relatively high correlation with the query. Otherwise, this correlation may be negative information. We incorporate the CWC module into the cross-attention block of the transformer to adjust the attention weights according to the consistent weighted correlations. Take the attention map obtained by multiplying the query and the key as input, and the new generated (q, k, v) is performed the attention again. The CWC module consistently adjusts the attention weights to strengthen the correct correlation between the relevant query—key pair and restrain the incorrect correlation between the irrelevant query–key pair. More specifically, the weights of the relevant query–key pairs are enhanced to strengthen the correct consistency, and the weights of the irrelevant query–key pairs are suppressed to alleviate the incorrect consistency. The CWC module computes the correlations of each query and all keys, and then the correlation scores are normalized. Finally, the attention weights are obtained by multiplying the normalized correlations and the consistent weights of the other query–key pairs.

By introducing the CWC module, we can consider the global context and consistency information in the attention mechanism to enhance correct correlations and suppress erroneous correlations. This can moderate the modeling capability for the correlation of the potential target and the surrounding disturbances, which facilitates the improvement in the behavior of the tracker. The experimental results indicate that the proposed CWCTrack can notably improve the tracking capability for both short- and long-term tracking benchmark tests, such as GOT-10K [13] and LaSOT [14].

2. The Framework of the Proposed Model

The framework of the proposed CWCTrack is shown in Figure 1.

As shown in Figure 1, the proposed CWCTrack is a Siamese network-based framework. The CWCTrack mainly contains three components: the feature extraction backbone network, the network for feature fusion (including the attention encoder–decoder), and the prediction

head. In the tracking process, making use of the shared weights, the features of image patches from the template and search region are extracted by the feature extraction network, considering the shared weights. The extracted features are merged into a feature sequence. Then, the concatenated feature sequences are sent to the encoder of the attention mechanism and enhanced layer-by-layer. The decoder network creates the final feature maps of the search regions. Finally, the feature maps are fed into the prediction head network to obtain the categorization response and the estimated bounding box.

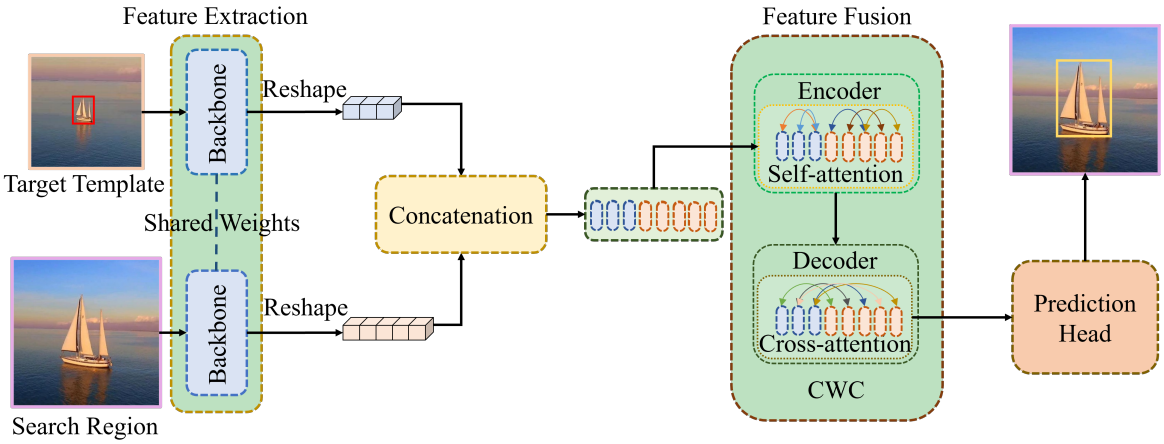


Figure 1. The framework of the proposed CWCTrack.

3. Methods

3.1. Backbone

Feature extraction plays a crucial role in the proposed CWCTrack framework. Similar to most of the transformer trackers [15,16], the starting frame with ground truth annotation is cropped as the template image patch ($x \in R^{3\times H_x\times W_x}$, *where* $H_x = W_x = 128$), which, together with the search region image patch ($z \in R^{3\times H_z\times W_z}$, *where* $H_z = W_z = 320$), are put into the network. For the extraction of template image patches, a specific area in the initial frame is selected according to the center coordinate of the potential target. The scope of this area is twice as long as the length of the local scene around the target. This template patch not only includes the appearance information of the target but also contains the local features of the target surroundings. On the other hand, the size of the search region patch is enlarged to a range of four times as long as the edge length of the central location of the target in the former frame, with the purpose of covering the potential movement of the target. To facilitate the following process, the template and search region patches are reconstructed into squares, from which the features are extracted by the feature extraction backbone network. Using this manipulation, we can obtain a regular feature representation suitable for the subsequent procedures, which is favorable for improving the accuracy of target tracking.

To facilitate our tracking task, an improved version of ResNet50 [17] is adopted. To maintain high feature resolution and capture more target detail information, the final step of ResNet50 is abandoned, and the outcome of the fourth step is employed as the ultimate feature map. Then, the 3×3 convolution of the fourth step is replaced by an expanded convolution with a step size of 2 for the purpose of enlarging the receptive field of the network. In this way, the perceptual range of features is expanded, enabling the network to better understand the feature representation of the search region and template. In order to further facilitate the resolution of features, we rectify the downsampling convolution step size from 2 to 1 in the fourth stage, thereby obtaining a more detailed feature map. Finally, the feature maps of the template and search region patches are obtained in the following form: $f_x \in R^{C\times H_{x'}\times W_{x'}}$ and $f_z \in R^{C\times H_{z'}\times W_{z'}}$, respectively, where $(H_{x'}, W_{x'}) = (H_x/s, W_x/s)$, $(H_{z'}, W_{z'}) = (H_z/s, W_z/s)$, $H_{x'} = W_{x'} = 8$, $H_{z'} = W_{z'} = 20$, $C = 1024$, and $S = 16$.

3.2. Encoder

Firstly, a 1×1 convolution is employed to obtain two low-dimensional feature maps of f_x and f_z, where the channel dimension is reduced from C (1024) to d (256). Then, we generate a feature sequence with length $L = H_{x'}W_{x'} + H_{z'}W_{z'}$ and dimension d by flattening the feature maps and connecting them along the spatial dimension, which is sent to the encoder of the transformer as the input. The transformer encoder includes N coding layers, and each layer involves a feedforward self-attention network with a multi-head block. With respect to the arrangement invariability of the prototype transformer [12], a sinusoidal positional embedding is combined into the input feature sequence. Finally, the encoder seizes the feature relationships among all the sequence components and uses global contextual information to enhance the original features, permitting the model to easily obtain distinguishing features of the target positioning.

3.3. Consistent Weighted Correlation (CWC) Module

In the transformer, the attention mechanism mainly consists of three components: query, key, and value. By performing a linear transform on the input sequence, a representation of the query, key, and value for each position is obtained. Query is used to specify the position we want to focus on, while key and value provide information about all positions in the sequence. Attention weight computing usually involves two steps: similarity computing between query and key, and normalization of the similarity. Common calculation methods include additive attention and dot product attention. In dot product attention, the inner product of query and key represents their similarity; in additive attention, the similarity is calculated via linear transform and the activation function processing of query and key. By calculating the attention weights, we can determine the importance of each position for the query. Then, we multiply and sum the attention weight with the corresponding position value to obtain the final context vector. This context vector contains weighted attention to different positions in the input sequence, which will be used for subsequent manipulations.

Using $Q, K, V \in R^{L \times d}$ to denote the matrix expression of query, key, and value, respectively, the attention module can be defined as follows:

$$Attention(Q, K, V) = (Softmax(\frac{\bar{Q}\bar{K}^T}{\sqrt{C}})\bar{V})W_o, \tag{1}$$

where $\bar{Q} = QW_q$, $\bar{K} = KW_k$, and $\bar{V} = VW_v$ represents different linear transform for Q, K, and V; and W_q, W_k, W_v, and W_o indicates the weight matrix of the linear transform.

As described in [12], by expanding the attention module to a multi-head way, the model is introduced into a multi-head attention module, which can capture the correlations and features from different aspects in a parallel way. This is beneficial for improving the modeling capability for the information in the input sequences. The multi-head attention mechanism provides a flexible way that permits the model to concentrate on different key value at the same time, which further enhances the expressive power and overall performance of the model. The multi-head attention module can be defined as

$$MultiHead(Q, K, V) = Concat(H_1, ..., H_{nh})W^o, \tag{2}$$

$$H_i = Attention(\bar{Q}\bar{W}_i^Q, \bar{K}\bar{W}_i^K, \bar{V}\bar{W}_i^V)W^o, \tag{3}$$

where $W^o \in R^{n_h d_v \times d_m}$, $\bar{W}_i^Q \in R^{d_m \times d_k}$, $\bar{W}_i^K \in R^{d_m \times d_k}$, and $\bar{W}_i^V \in R^{d_m \times d_k}$ is the parameter matrix, respectively.

For the typical attention mechanism, the relationship between query and key is independently computed in the feature association mapping $N = \frac{\bar{Q}\bar{K}^T}{\sqrt{C}} \in R^{L \times L}$, neglecting the connections with other potential query–key pairs. This will deteriorate the informa-

tion propagation in cross-attention and diminish the identification performance of the transformer tracker.

To better understand the importance of different pieces of the input information, a consistent weighted correlation (CWC) module is proposed to compute the correlation between the query–key pair, which sustains the flexibility of attention weights. By introducing the CWC module, the correct correlations between the relevant pairs are strengthened and the incorrect correlations between the irrelevant pairs are suppressed. Both the feature aggregation and the information propagation are improved when the erroneous correlations are eliminated. This improvement is beneficial to the precision of the attention, thus greatly promote the tracking ability of the CWCTrack, especially for the complex scenarios.

Specifically, we refine the feature association mapping $N = \frac{\bar{Q}\bar{K}^T}{\sqrt{C}} \in R^{L\times L}$ in the cross-attention prior to the softmax step, as Figure 2 illustrates. We treat the columns in N as a correlation vector sequence, and the internal attention block outputs a residual correlation map using these columns as query $Q^{'}$, key $K^{'}$, and value $V^{'}$. Considering the input matrix $Q^{'}$, $K^{'}$, and $V^{'}$, we first obtain the transformed version of query and key, i.e., $\bar{Q}^{'}$ and $\bar{K}^{'}$, as shown in the left part of Figure 2. More specifically, the scale of $Q^{'}$ and $K^{'}$ is reduced to $L \times d$ $(d \ll L)$, for the purpose of increasing the computational efficiency. After normalization [18], a 2-D sinusoidal encoding [19] is added to supply position clues. Furthermore, the normalized version of value $V^{'}$ is produced by a normalization operation, i.e., $\bar{V}^{'} = LayerNorm(V^{'})$. Finally, a residual correlation map of the normalized version $\bar{Q}^{'}$, $\bar{K}^{'}$, and $\bar{V}^{'}$ is derived by the internal attention module via the following equation:

$$InnerAttn(N) = (Softmax(\frac{\bar{Q}^{'}\bar{K}^{'T}}{\sqrt{D}})\bar{V}^{'})(1 + W_o^{'}), \tag{4}$$

where $W_o^{'}$ is the weights of linear transform used to adjust the aggregated correlations under an identical connection.

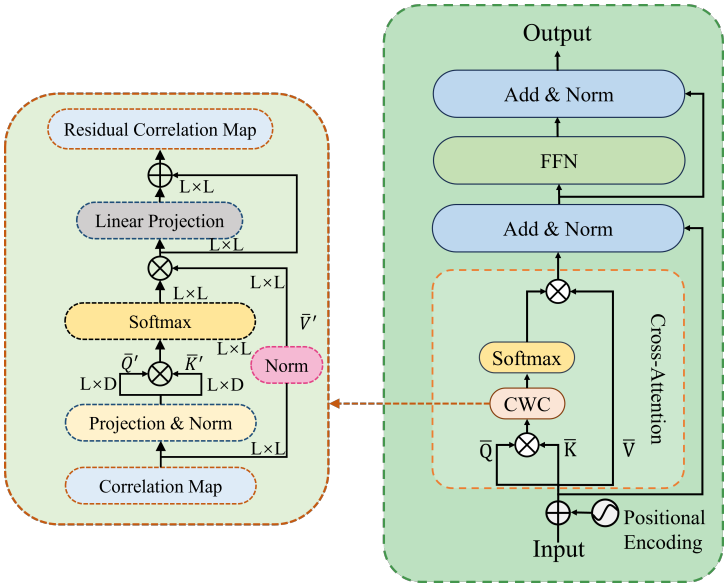


Figure 2. The details of CWC module (**left**) and the example of cross-attention module (**right**). The CWC module performs similarity calculation (correlation map) by inserting it into the query and the key. Add & Norm represents residual structure and normalization. FFN represents feedforward network. Softmax is an activation function. Residual Correlation Map represents the reconstructed similarity.

Intrinsically, the CWC module produces the residual correlation vector for each correlation vector of the correlation map N by aggregating the original correlation vectors. Using

this operation, we can explore the consensus between the correlations in the receptive field. The proposed CWC attention block can be formulated by the residual correlation map as

$$CorrAttn(Q, K, V) = (Softmax(N + InnerAttn(N))\bar{V})W_o. \tag{5}$$

The CWC module enables multiple attention heads in parallel to share the same weighting parameters, which can decrease the count of parameters in the model and improve its efficiency. This way, the complexity of the tracking model is greatly reduced, while still maintaining good performance.

3.4. Prediction Head

In the conventional TransT, the prediction head is designed by the ordinary MLP (Multi-Layer Perceptron) and a ReLU (Rectified Linear Unit) activation function [7]. However, this kind of design is neither flexible nor robust to many challenges in tracking tasks, such as occlusion, background clutter, etc. To improve the positioning accuracy of the tracking box, we employ the probability distribution prediction head of box estimation in the STARK [16]. Firstly, the feature maps of the search region are picked up from the output sequences of the Decoder. Then, the similarity of feature between the search region and the embeddings of the Encoder output is computed. Secondly, to obtain enhanced features, the search region features are multiplied by the similarity scores element-wisely, which can strengthen the important areas and suppress the non-discriminative areas. The enhanced feature sequences are reorganized to a feature map $f \in R^{d \times H_{z'} \times W_{z'}}$, which is sent to a simple FCN (Fully Convolutional Network). FCN comprises L piled layers of Conv-BN-ReLU, from which the probability maps of the upper left and lower right corners of the object bounding box $P_{tl}(x, y)$ and $P_{br}(x, y)$ are produced separately. In the end, the coordinates of the potential bounding box $(\bar{x}_{tl}, \bar{y}_{tl})$ and $(\bar{x}_{br}, \bar{y}_{br})$ are obtained by computing the expectations of the corner probability distribution via the following equation:

$$\begin{cases} (\bar{x}_{tl}, \bar{y}_{tl}) = (\sum_{y=0}^{H} \sum_{x=0}^{W} x \cdot P_{tl}(x, y), \sum_{y=0}^{H} \sum_{x=0}^{W} y \cdot P_{tl}(x, y)), \\ (\bar{x}_{br}, \bar{y}_{br}) = (\sum_{y=0}^{H} \sum_{x=0}^{W} x \cdot P_{br}(x, y), \sum_{y=0}^{H} \sum_{x=0}^{W} y \cdot P_{br}(x, y)). \end{cases} \tag{6}$$

3.5. Loss Function for Training

The prediction head takes over the feature sequences and yields a classification result of binary regression (both the input and output are with size of $H_{z'} W_{z'}$). The feature sequences that correspond to the pixels located in the realistic bounding box are chosen to be positive subsets, and the rest are categorized as negative subsets. All elements of the feature participate in the computation for the classification loss, whereas only the positive subsets participate in the computation for regression loss. To alleviate the instability between the positive and negative subsets, we downgrade the loss caused by the negative subsets to 1/16. Finally, the classification loss adopting canonical binary cross entropy is formulated as follows:

$$L_{cls} = -\Sigma_j [y_j log(p_j) + (1 - y_j) log(1 - p_j)], \tag{7}$$

where y_j is the authentic label of the j-th component ($y_j = 1$ indicates the foreground), and p_j is the probability when the learned model concludes that the prediction belongs to the foreground. The regression loss comprises the linear weighted loss of L_1-norm and L_{GIoU} [20], which is formulated by

$$L_{reg} = \Sigma_j [\lambda_1 L_1(b_j, \hat{b}) + \lambda_g L_{GIoU}(b_j, \hat{b})], \tag{8}$$

where L_1 and L_{GIoU} represent the L_1 loss and the generalized IoU loss, respectively. $\lambda_1 L_1$ and $\lambda_g L_{GIoU}$ are the hyperparameters determining the relative impact of the two kinds of loss functions. b_j is the j-th predictive bounding box and $\hat{b}$ is the normalization of the true bounding box. In our implementation, the regularization parameters λ_g and λ_1 are set as 2 and 5, respectively.

4. Experimental Results

In this section, we first describe the conduction details of the proposed CWCTrack conducted on several prevalent tracking benchmarks. Then, the tracking results of the CWCTrack are depicted and compared with some of the most advanced trackers. Furthermore, we carry out ablation tests to validate the contribution of each component. In the end, we visualize the tracking results of four typical sequences from the OTB100 dataset [21].

4.1. Implementation Details

The proposed CWCTrack is conducted using Python 3.7 and PyTorch 1.13.0, and the tracking experiments are implemented on a NVIDIA GeForce RTX 4090 server. The training data includes GOT-10K [13], LaSOT [14], COCO2017 [22], and TrackingNet [23]. The patch size of the template and search region is set to 128×128 and 320×320, respectively, and the selected box areas of the template and search region are 2 and 4 times enlarged from the center of the target, respectively. In addition, data augmentations are also employed, including horizontal flipping and brightness jitter. CWCTrack uses ResNet50 [17] as the backbone and initializes the backbone with pre trained parameters on ImageNet. The BatchNorm [24] layer was frozen during training with six encoder and six decoder layers, consisting of multi-head attention layers (MHA) and feedforward networks (FFN). MHA has eight heads (with $width = 256$), while the FFN have hidden units of 2048. The dropout ratio value is 0.1. The bounding box prediction head is a lightweight FCN, consisting of five stacked Conv-BN-ReLU layers. The classification head is a three-layer perceptron with 256 hidden units in each layer. The CWCtrack completely trained 500 epochs, and after 400 epochs, the learning rate is downshifted by a factor of 10. The initial learning rates of the backbone and the rest parts are 10^{-5} and 10^{-4}, respectively. The network is optimized using the AdamW optimizer [25] with a weight decay of 10^{-4}.

4.2. Results and Comparisons

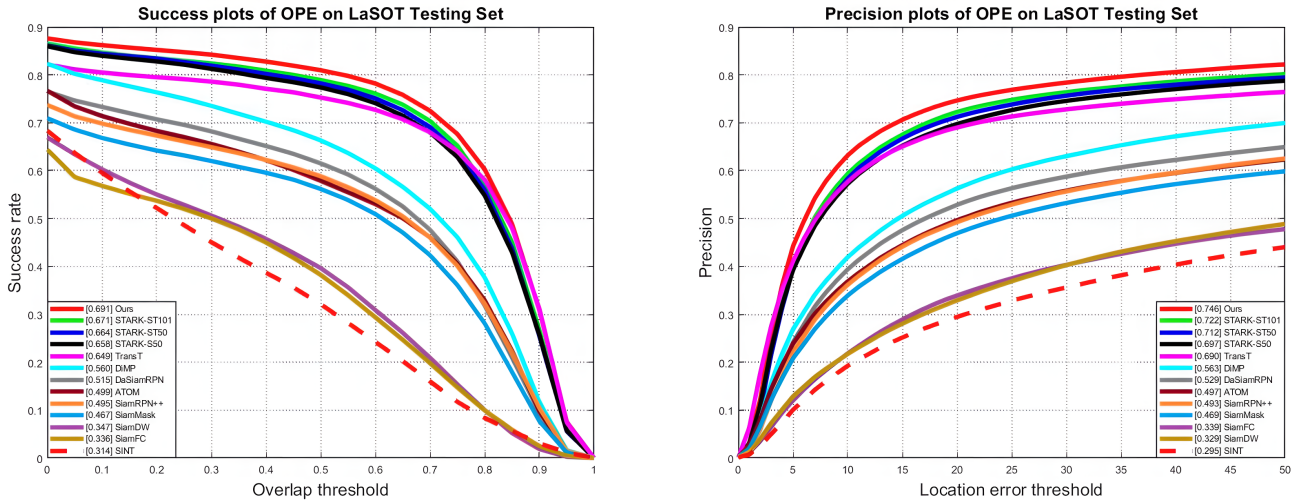
We validate the proposed CWCTrack with four commonly used datasets, including the online object tracking benchmark dataset OTB100 [21] and three large-scale bench-mark test datasets GOT-10K [13], LaSOT [14], and UAV123 [26].

GOT-10K includes over 10,000 video sequences for moving objects in reality, with more than 1.5 million handmade bounding boxes, which provides enough scenarios for large-scale target tracking benchmarks. It covers various challenges such as fast-moving objects, large-scale changes, cluttered backgrounds, occlusions, etc. It requires the tracker to only use the training set for model learning. Following this, we retrain the proposed CWCTrack model only using the training set of GOT-10K. The tracking results are summarized in Table 1. As we can see, the proposed approach gains an advantage over the former best tracker STARK-S50 [16] by 1.6% for the AO score. Furthermore, the proposed approach outperforms STARK-S50 by 0.4% for the SR0.75 score. For the SR0.5 score, the proposed approach has also achieved very close performance compared with the best tracker TransT [7].

Table 1. Comparisons of the tracking results on GOT-10K.

	SiamFC [5]	**ATOM [27]**	**Ocean [28]**	**STARK-S50 [16]**	**TransT [7]**	**Ours**
AO (%)	34.8	55.6	61.1	67.2	67.1	**68.8**
SR0.5 (%)	35.3	63.4	72.1	76.1	**76.8**	**76.4**
SR0.75 (%)	9.8	40.2	4.3	61.2	60.9	**61.6**

LaSOT provides a long-term single object tracking benchmark, which comprises 1550 carefully annotated video sequences with over 3.87 million frames. The tracking results are depicted in Figure 3. Our method is compared with different variants of STARK [16], TransT [7], DiMP [29], DaSiamRPN [30], ATOM [27], SiamMask [31], SiamDW [32], and SINT [33]. As can be seen, our approach outperforms the other competitive trackers. More precisely, the CWCTrack achieves the highest AUC (area-under-the-curve) score (i.e., success rate) of 69.1%, which is 2% higher than the former best tracker STARK-ST101, as shown in Figure 3a. For precision plotting, the proposed CWCTrack also achieves the highest score of 74.6%, 2.4% higher than STARK-ST101, as shown in Figure 3b. It should be noted that the results of the STARK-ST101 are reproduced by our own manipulation, which will inevitably deviate from the original performance reported by the author.



(**a**) (**b**)

Figure 3. Comparisons of the tracking results on LaSOT dataset. (**a**) Success rate; (**b**) precision.

Figure 4 describes a comprehensive exhibition of the tracking results with various scenarios. The proposed CWCTrack attains excellent performances for all scenarios, especially for background clutter, fast motion, full occlusion, illumination variation, and low resolution. Table 2 shows the quantitative AUC results from Figure 4 and the precision results. As can be seen, for both the AUC score and precision, our proposed method achieves the best performance.

For intuitive comparison, the radar chart in Figure 5 provides an attribute-based assessment of the tracking results. Our approach succeeds in most of the attribute partitions, which implies the feasibility and validity of the proposed model.

OTB100 contains a total of 100 sequences with each frame annotated. It introduces 11 challenge attributes for performance analysis. Figure 6 shows the comparisons of the proposed approach with three state-of-the-art trackers; as one can observe, the proposed approach attains almost equivalent or even superior performance compared to the reference models. For the success rate, the CWCtrack is 0.3% and 6.2% higher than the transformer-based trackers Transt [7] and TCTrack [34], respectively. (The TCTrack is suitable for drone tracking and may not perform well on small datasets such as OTB100).

UAV123 contains 123 video sequences from the aerial viewpoint. Evaluated by the success rate and accuracy, respectively, the tracking results of various trackers are shown in Table 3. As can be observed, the proposed approach attains better performance in contrast to the competitors both in AUC score (success rate) and in precision.

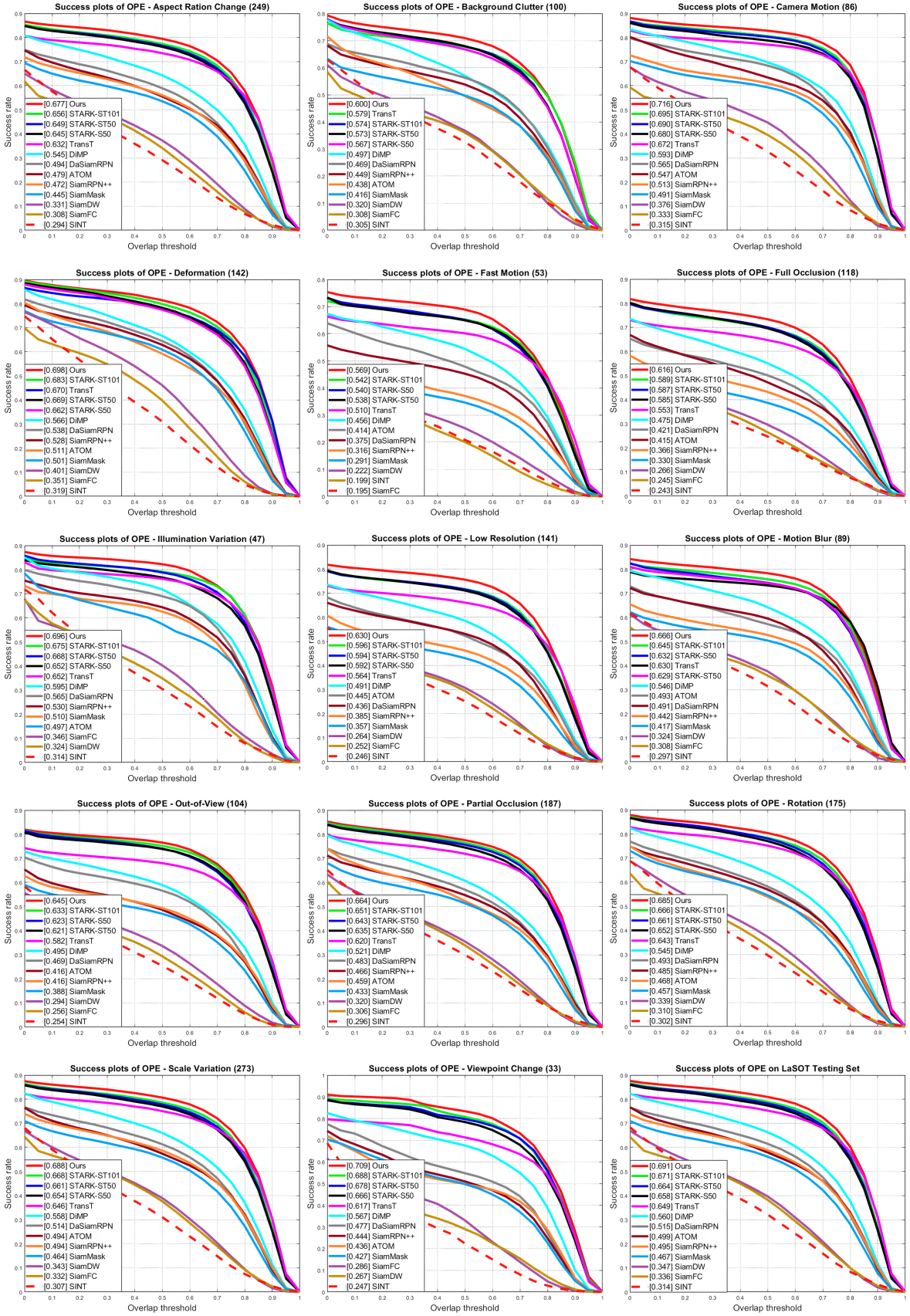


Figure 4. The AUC results of the LaSOT dataset under different challenge scenarios. The figures are best viewed by zooming in. The raw data and high-resolution figures are available upon request.

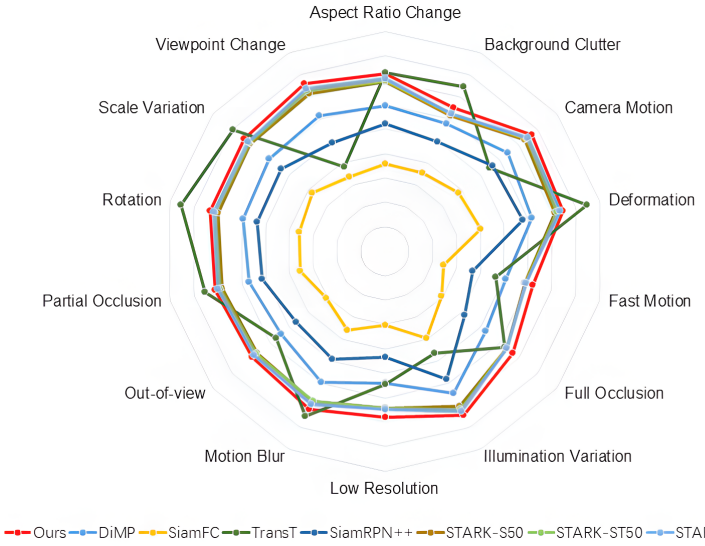


Figure 5. Radar chart for attribute-based assessment of the trackers on LaSOT for AUC score.

Table 2. Comparisons of the LaSOT dataset.

	AUC (%)	Precision (%)
Ours	**69.1**	**74.6**
STARK-101	67.1	72.2
STARK-ST50	66.4	71.2
STARK-S50	65.8	69.7
TransT	64.9	69.0
DiMP	56.0	56.3
DaSiamRPN	51.5	52.9
ATOM	49.9	49.7
SiamMask	49.5	46.9
SiamDW	46.7	32.9
SiamFC	34.7	33.9
SINT	31.4	29.5

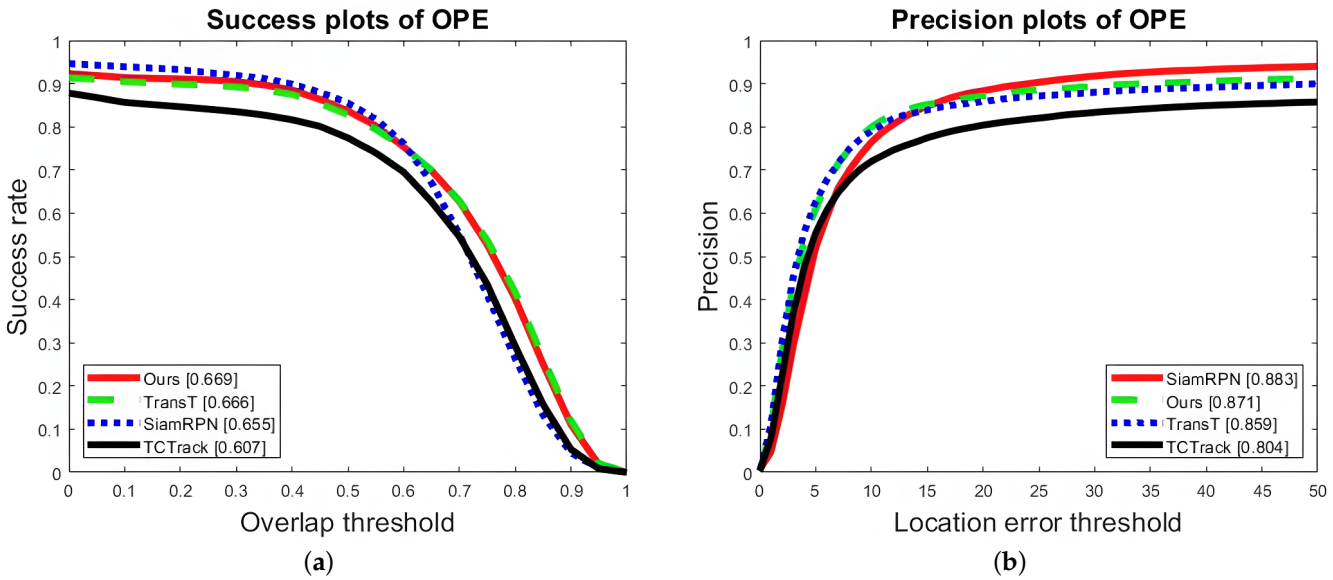


Figure 6. Comparisons of the tracking results on OTB100 dataset. (**a**) Success rate; (**b**) precision.

Table 3. Comparisons of the tracking results on UAV123.

	SiamFC [5]	ATOM [27]	Ocean [28]	TransT [7]	Ours
AUC (%)	49.2	61.7	62.1	68.1	**68.2**
Precision (%)	72.7	82.7	82.3	87.6	**88.3**

4.3. Ablation Analysis

To examine the significance of each constituent part in the proposed CWCTrack, ablation tests are executed on the testing set of LaSOT. The ablation experimental results are illustrated in Table 4. For simplicity, the encoder, decoder, consistent weighted correlation module, and position coding is abbreviated by Enc, Dec, CWC, and Pos, respectively. The blank indicates the component is adopted by default; on the other hand, ⊗ indicates that the component has been deleted. #1 indicates that when the encoder is erased from the tracker, the success rate is reduced by 5.9%. This indicates that the intensive interaction between template features and search regions plays a crucial role for the tracking task. When the decoder is erased, the success rate decreases by 3.7%, as shown by #2. This decrease is much less than that of erasing the encoder, indicating that the encoder is of more important significance than the decoder. When we delete the CWC module, the success rate decreases by 2.7%, indicating that the CWC module facilitates the attention of the decoder to some extent, as shown by #3. Finally, as shown by #4, the success rate only decreases by 0.4% when the position coding is removed, so we can conclude that the position coding is not as important as the other components in the proposed tracker.

Table 4. Ablation tests on LaSOT.

#	Enc	Dec	CWC	Pos	Success (%)
1	⊗				63.2
2		⊗			65.4
3			⊗		66.4
4				⊗	68.7
5					69.1

4.4. Visualization of the Tracking Results

To evaluate the validity of the proposed CWCTrack, we depict some tracking results conducted on OTB100 dataset in Figure 7, together with three other representative trackers. As can be seen, the tracking results of CWCTeack conducted on four typical video sequences surpass that of the other trackers.

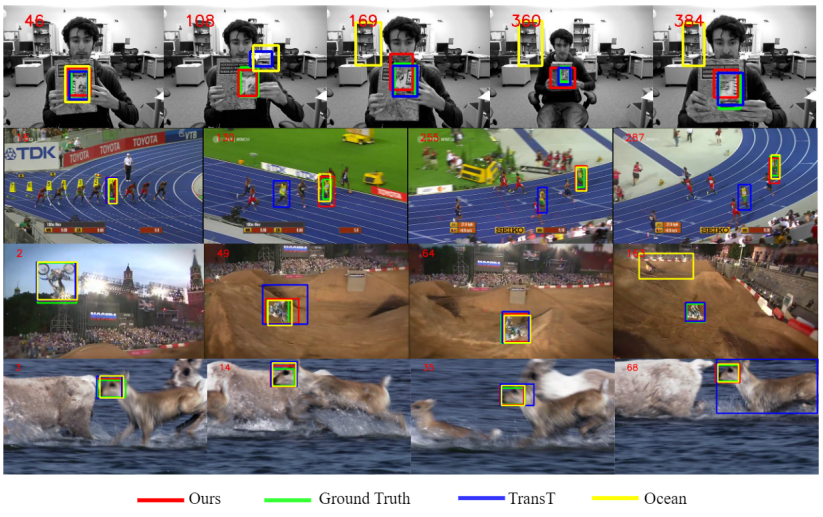


Figure 7. Visualization of the tracking results on four sequences from the OTB100 dataset.

5. Discussions

In the context of our proposed CWCTrack framework for transformer-based visual tracking, our experimental results on four widely recognized benchmarks have provided valuable insights and demonstrated promising outcomes. Our results indicate that incorporating the CWC module into the cross-attention block significantly improves the performance of transformer-based visual tracking. The CWC module addresses the issue of neglecting correlations between queries and keys, resulting in more accurate attention mechanisms. This finding aligns with the importance of attention mechanisms in visual tracking, as demonstrated by previous studies. Our approach provides a novel solution to enhance feature representation in both the search and template regions, contributing to better tracking accuracy.

The implications of our work can be extended beyond the specific task of visual tracking. Attention mechanisms are fundamental in various fields, including natural language processing and computer vision. Our proposed CWCTrack framework highlights the potential of attention mechanisms to be further fine-tuned and adapted to specific application domains, enhancing their robustness and accuracy. This suggests that our research can inspire advancements not only in visual tracking but also in other domains where attention mechanisms are applied.

It is essential to notice the limitations of our study; while CWCTrack demonstrates promising results, it is not without constraints. One limitation is that our approach may require additional computational resources due to the additional complexity of the CWC module. Furthermore, the generalization of our framework across various tracking scenarios and datasets needs further investigation. Moreover, we recognize that the performance improvement may not be substantial in all cases.

Future work includes how to optimize and speed up the CWC module to improve the real-time performance of the model. First, it is crucial to optimize the computational efficiency of the CWC module without compromising the tracking accuracy, which makes it more practical for real-time applications. Second, exploring the adaptability of CWCTrack to different tracking scenarios and datasets can help uncover its full potential abilities. Furthermore, there is room for exploring hybrid models that combine attention mechanisms with other techniques to further enhance tracking performance. Finally, investigating the transferability of the CWC module to other computer vision tasks beyond tracking is an intriguing direction.

6. Conclusions

This paper introduces a consistent weighted correlation (CWC) module to refine the attention mechanism, which is crucial in transformer-based visual tracking. By inserting the CWC module into the cross-attention block of the transformer, we eliminated the issue of the independent computing of the correlations in existing methods. The consistent principle is adopted to enhance the correct correlations and suppress the erroneous correlations. By considering the global context and consistent information, the CWC module can capture the correlations between the object and surroundings more accurately and improve the distinguishing capability of the model for the relationship between the target and the disturbance. Conducted on four popular tracking benchmarks, the tracking results reveal that the proposed CWCTrack attains promising performance compared to the state-of-the-art tracking models.

Author Contributions: Conceptualization, L.L. and G.F.; methodology, G.F.; software, G.F.; validation, J.W.; writing—original draft preparation, G.F. and J.W.; writing—review and editing, L.L. and S.N.M.; supervision, L.S.; project administration, K.Z.; funding acquisition, S.W. and C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 62006092); the Natural Science Foundation of Hebei Province (No. F2022201013); the Scientific Research Program of the Anhui Provincial Ministry of Education (No. KJ2021A0528 and KJ2020A1202);

the University Synergy Innovation Program of Anhui Province (GXXT-2022-033); the Start-up Foundation for Advanced Talents of Hebei University (No. 521100221003); the Laboratory Opening Project of CHNU (No. 2022sykf046); and Anhui Shenhua Meat Products Co., Ltd. Cooperation Project (No. 22100084).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are available from public datasets.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marvasti-Zadeh, S.M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep learning for visual tracking: A comprehensive survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 3943–3968. [CrossRef]
2. Fiaz, M.; Mahmood, A.; Javed, S.; Jung, S.K. Handcrafted and deep trackers: Recent visual object tracking approaches and trends. *ACM Comput. Surv.* **2019**, *52*, 1–44. [CrossRef]
3. Wang, Q.; Teng, Z.; Xing, J.; Gao, J.; Hu, W.; Maybank, S. Learning attentions: Residual attentional siamese network for high performance online visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4854–4863.
4. Du, F.; Liu, P.; Zhao, W.; Tang, X. Correlation-guided attention for corner detection based visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6836–6845.
5. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision Workshops, Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.
6. Chen, B.; Li, P.; Bai, L.; Qiao, L.; Shen, Q.; Li, B.; Ouyang, W. Backbone is all your need: A simplified architecture for visual object tracking. In Proceedings of the European Conference on Computer Vision, Tel-Aviv, Israel, 23–27 October 2022; pp. 375–392.
7. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8126–8135.
8. Chen, H.; Wang, Z.; Tian, H.; Yuan, L.; Wang, X.; Leng, P. A Robust Visual Tracking Method Based on Reconstruction Patch Transformer Tracking. *Sensors* **2022**, *22*, 6558. [CrossRef] [PubMed]
9. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. Mixformer: End-to-end tracking with iterative mixed attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13608–13618.
10. Ye, B.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Joint feature learning and relation modeling for tracking: A one-stream framework. In Proceedings of the European Conference on Computer Vision, Tel-Aviv, Israel, 23–27 October 2022; pp. 341–357.
11. Zhou, J.; Yao, Y.; Yang, R.; Xia, Y. D-TransT: Deformable Transformer Tracking. *Electronics* **2022**, *11*, 3843. [CrossRef]
12. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
13. Huang, L.; Zhao, X.; Huang, K. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1562–1577. [CrossRef] [PubMed]
14. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5369–5378.
15. Chen, X.; Yan, B.; Zhu, J.; Lu, H.; Ruan, X.; Wang, D. High-performance transformer tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 8507–8523. [CrossRef] [PubMed]
16. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning spatio-temporal transformer for visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10448–10457.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
19. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
20. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
21. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
22. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.

23. Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 310–327.
24. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
25. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
26. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for UAV tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
27. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate tracking by overlap maximization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4655–4664.
28. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-aware anchor-free tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 771–787.
29. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Learning discriminative model prediction for tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6182–6191.
30. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 101–117.
31. Hu, W.; Wang, Q.; Zhang, L.; Bertinetto, L.; Torr, P.H. Siammask: A framework for fast online object tracking and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3072–3089. [PubMed]
32. Zhang, Z.; Peng, H. Deeper and wider siamese networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4591–4600.
33. Tao, R.; Gavves, E.; Smeulders, A.W. Siamese instance search for tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429.
34. Cao, Z.; Huang, Z.; Pan, L.; Zhang, S.; Liu, Z.; Fu, C. TCTrack: Temporal contexts for aerial tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14778–14788.