electronics

Article

FQTrack:Object Tracking Method Based on a Feature-Enhanced Memory Network and Memory Quality Selection Mechanism

Jianwei Zhang [1], Mengya Zhang [1,*], Huanlong Zhang [2], Zengyu Cai [3] and Liang Zhu [3]

1 School of Software, Zhengzhou University of Light Industry, Zhengzhou 450000, China; ing@zzuli.edu.cn
2 School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China; hlzhang@zzuli.edu.cn
3 School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China; czy@zzuli.edu.cn (Z.C.); lzhu@zzuli.edu.cn (L.Z.)
* Correspondence: 332113020729@email.zzuli.edu.cn

Abstract: Visual object tracking technology is widely used in intelligent security, automatic driving and other fields, and also plays an important role in frontier fields such as human–computer interactions and virtual reality. The memory network improves the stability and accuracy of tracking by using historical frame information to assist in the positioning of the current frame in object tracking. However, the memory network is still insufficient in feature mining and the accuracy and robustness of the model may be reduced when using noisy observation samples to update it. In view of the above problems, we propose a new tracking framework, which uses the attention mechanism to establish a feature-enhanced memory network and combines cross-attention to aggregate the spatial and temporal context information of the target. The former introduces spatio-temporal adaptive attention and cross-spatial attention, embeds spatial location information into channels, realizes multi-scale feature fusion, dynamically emphasizes target location information, and obtains richer feature maps. The latter guides the tracker to focus on the area with the largest amount of information in the current frame to better distinguish the foreground and background. In addition, through the memory quality selection mechanism, the accuracy and richness of the feature samples are improved, thereby enhancing the adaptability and discrimination ability of the tracking model. Experiments on benchmark test sets such as OTB2015, TrackingNet, GOT-10k, LaSOT and UAV 123 show that this method achieves comparable performance with advanced trackers.

Keywords: object tracking; siamese network; memory network; attention mechanism; spatial temporal context

check for updates

Citation: Zhang, J.; Zhang, M.; Zhang, H.; Cai, Z.; Zhu, L. FQTrack:Object Tracking Method Based on a Feature-Enhanced Memory Network and Memory Quality Selection Mechanism. *Electronics* **2024**, *13*, 3221. https://doi.org/10.3390/electronics13163221

Academic Editor: Manohar Das

Received: 6 June 2024
Revised: 22 July 2024
Accepted: 31 July 2024
Published: 14 August 2024

1. Introduction

Object tracking is a key task in the field of computer vision, which is widely used in sports event broadcasting, security monitoring, automatic driving, and robotics [1,2]. Target tracking primarily involves detecting, localizing, and tracking the position and trajectory of target objects across consecutive images or video sequences. In recent years, despite the significant progress made in the field of target tracking, challenges such as changes in target size, interference factors, and occlusion phenomena still lead to a certain gap between theory and practical application [3,4].

In recent years, most popular object-tracking frameworks based on Siamese networks primarily rely on target appearance information from the initial frame for tracking. However, the appearance of the target continues to change during movement. Relying solely on the static appearance from the initial frame lacks dynamic perception of the target information, making it difficult to handle appearance changes caused by occlusion and non-rigid deformation. Therefore, effectively remembering and extracting target information is a significant challenge in object tracking. To address this challenge, trackers should comprehensively utilize multi-frame historical appearance information to construct adaptive models, rather than relying solely on previous or initial frames. Based on this, object-tracking methods based on memory networks are gradually emerging. Some trackers utilize online updating techniques [5–10] to update the model using historical frames. However, this approach requires customized updating mechanisms, which may increase the complexity of the model and reduce tracking efficiency. On the other hand, some trackers [11–13] utilize features collected from multiple historical target states during the tracking process to generate adaptive templates for target modeling. For instance, in the realm of tracking algorithms, UpdateNet [11] innovatively introduced a template update network mechanism. This mechanism combines the initial template, accumulated template information from the preceding frame, and the current frame template to anticipate and refine the tracking template for the subsequent frame, markedly improving the efficiency and precision of template updating. In contrast, the methodology proposed by Stark [12] emphasizes the intelligent fusion of the initial frame template, current frame data, and dynamically adjusted templates. Updates to the template are initiated solely when the confidence score, as determined by this system, surpasses a predefined threshold. MixFormer [13] designed a score prediction module and selected reliable online templates for updating according to the predicted scores.

Although the memory mechanism has achieved some success in visual object tracking, it still has limitations in complex environments. First of all, these methods [5,11] only obtain dynamic adaptive templates by updating mechanisms, ignoring the mining and enhancement of effective features. Secondly, some trackers [12,14,15] use the preset threshold to select memory frames, which increases the computational complexity when the number of frames satisfies the condition is large. Although these mechanisms improve the robustness of the tracker, they inevitably introduce noise templates that make it difficult to effectively distinguish between the background and foreground, resulting in tracking drift. In response to these limitations, three problems are proposed: (1) how to obtain more diverse and high-quality memory frames and reduce noise samples; (2) how to improve the feature expression ability of memory frames and dig deep into the effective information; and (3) how to make full use of time and space context information.

This paper proposes an innovative target-tracking strategy, which combines feature-enhanced memory network technology and memory quality selection mechanisms to improve the accuracy and robustness of the tracking process. By enhancing the feature representation ability of the target memory frame and obtaining a high-quality and diversified target appearance feature set, the method further improves the tracker's ability to distinguish between the target and the background, there by enhancing the tracker's adaptive ability and efficient discrimination ability in a dynamic environment. The main contributions of this paper are as follows:

- A feature-enhanced memory network is designed to enhance and mine rich spatial representation of the target through the attention mechanism and aggregate temporal and spatial context information through cross-attention so that the tracker can better locate and distinguish the target;
- The online memory quality selection mechanism is proposed and the quality evaluator is used to judge whether the current frame is suitable for updating memory and reducing noise interference. For appearance changes, the mechanism can select a variety of memory frames to enrich the sample set and enhance the adaptive ability of the tracker;
- The proposed method performs well compared with some advanced methods on benchmark datasets such as OTB2015, TrackingNet, GOT-10k, LaSOT and UAV 123, demonstrating the effectiveness of the proposed method.

2. Related Work

2.1. Siamese-Based Trackers

The development of deep learning has greatly promoted the progress of object-tracking technology. The traditional correlation filtering algorithm is fast but the tracking quality in complex scenes is not good. After the introduction of deep neural networks, object tracking significantly improves the accuracy and robustness by learning deep features. MDNet [14] achieves leading performance by pre-training deep neural networks to learn target representation and adapting to specific tracking tasks through online fine-tuning. Deep learning methods have demonstrated remarkable performance in object tracking, yet their high computational complexity and relatively slow speed limit their widespread adoption in real-time applications. To address this challenge, SiamFC [16], a pioneer in the field of Siamese network tracking, employs a lightweight network architecture that parallelly processes template and search images while computing their similarity for fast matching. This design not only simplifies the tracking process but also maintains high performance, fulfilling the stringent requirements of real-time tracking. As a result, SiamFC has paved a new way for research in real-time object tracking. Subsequently, object tracking methods based on Siamese networks [17–25], such as SiamRPN [17] and SiamRPN++ [18], have introduced region proposal networks and deeper structures to improve the accuracy and robustness of tracking. The tracking method based on the Siamese network solves the problem of template matching but some trackers have limited adaptive ability and are difficult to adapt to the changes in the target in the video sequence. In contrast, we model the target by using multiple historical frame information, fully exploit the context information of the target, and improve the adaptive ability of the tracker.

2.2. Memory Networks-Based Trackers

The core goal of a memory network [26,27] is to overcome the inherent limitations of traditional neural network models in dealing with long-term memory and complex reasoning tasks (such as automatic question-answering, dialogue systems, etc.). By introducing an external memory mechanism, the memory network enables the model to store and effectively use historical information, thus showing higher performance when dealing with such tasks. Its flexible structure can be adjusted and optimized according to the task, so it is applied to object tracking and other fields. With the deepening of research, more and more work has begun to explore how to effectively use memory networks to improve performance in object-tracking tasks. MemTrack [28] is a classic tracker, which introduces LSTM to control the update of the template. Because LSTM uses additional storage space, the storage space is more flexible. However, the accuracy of the tracker is not high, and it is easy to cause memory loss when performing memory reading. MLT [29] applies the idea of deep meta-learning to target tracking. This method can use historical frame information and generate adaptive update weights based on existing tracking templates to achieve dynamic updates of templates. Zhang et al. proposed a lightweight regression memory network [30], which adopts a single convolutional layer structure, learns and stores the target appearance memory through convolutional linear regression, and adjusts the parameters according to the reliable tracking results of each frame to strengthen the memory of the target appearance changes. In order to make better use of the time and space information in the target historical frame, STEM [31] developed a memory model of the key–value structure. The model improves the adaptive ability and distinguishing ability of the tracker by mining the key–value relationship between the target and the query. Although these trackers, based on memory networks, improve the adaptive ability, they have limitations in the selection of memory frame quality and diversity, have a large amount of calculation, and are susceptible to noise interference. Differently, we use the online memory quality selection mechanism to screen out high-quality and diverse memory frames for tracking, which helps to reduce noise interference, reduce error accumulation, and respond more effectively to target appearance changes.

2.3. Attention-Based Trackers

The attention mechanism holds a pivotal position in the field of object tracking. Specifically, by dynamically adjusting the focus on different regions within an image or video frame, the attention mechanism enables tracking algorithms to adaptively identify and concentrate on the features that are crucial for maintaining stable tracking. This process not only minimizes the waste of computational resources but also significantly enhances the ability of tracking systems to handle complex scene variations, such as illumination changes, occlusions, and rapid movements, thereby reducing the risk of tracking failures. The CSR-DCF [32] algorithm introduces channel and spatial attention mechanisms and trains and weights the final response map through the foreground space graph constraint filter. SiamGAT [33] introduced a novel target perception strategy into the Siamese network framework and realized the interaction between template features and search features by cleverly integrating the graph attention mechanism. RASNet [34] skillfully combines three attention mechanisms to weigh the features in the SiamFC framework, which significantly improves the discriminative ability of the tracker. EDCF [35] uses an autoencoder network to focus on target details, combined with context-aware correlation filtering and online updates. The introduction of a transformer [36] brings global modeling capabilities to object tracking, improving accuracy and stability. TransT [37] proposed a feature fusion model based on a transformer, which significantly improves the accuracy of the algorithm by establishing nonlinear semantic fusion and mining global information of long-distance feature association to effectively aggregate targets and search regions. The PrDiMP [38] algorithm innovatively integrates the transformer architecture into the field of target tracking and constructs a powerful sequence processing framework, which can make full use of the rich information of historical frames to enhance the feature expression of the current search area. This design not only breaks the limitations of traditional tracking methods in feature extraction but also provides a more comprehensive context-aware capability for the tracker through the fusion of cross-frame information. While our tracker uses a transformer-based cross-attention mechanism to improve adaptive ability and discrimination ability. The study of the above tracker shows that the careful design of the attention mechanism can improve the accuracy and robustness of the tracking algorithm and verify its effectiveness. Different from this, our tracker adopts a transformer-based cross-attention mechanism, which uses historical frames as input to realize the aggregation of temporal and spatial context information, and enhances the adaptive ability and discrimination ability of the tracker.

3. Proposed Method

3.1. Overall Architecture

As shown in Figure 1 the tracking framework architecture consists of four key modules: the feature extraction network module, feature enhancement module, memory quality selection module, and head network module. The feature extraction network module is divided into a feature enhancement branch (upper branch) and a search branch (lower branch). The former deals with a mask-optimized initial template or memory frame and the latter is used for feature extraction of search frames.

In the tracking initialization phase, the template frame is first preprocessed by the mask and then enters the feature extraction network to extract its key features. These features are further enhanced by the feature-enhanced attention mechanism to generate enhanced template features, which are stored in the memory storage to provide a reference for subsequent frame tracking. Then, the search branch extracts the features of the search frame and selects the memory frame features from the memory storage to connect them in series. The concatenated features and the search frame features are jointly input into the cross-attention mechanism to generate a fusion feature map with rich context information. Then, the fusion feature map is classified and regressed through the head network and, finally, the tracking result map with the target bounding box is output.

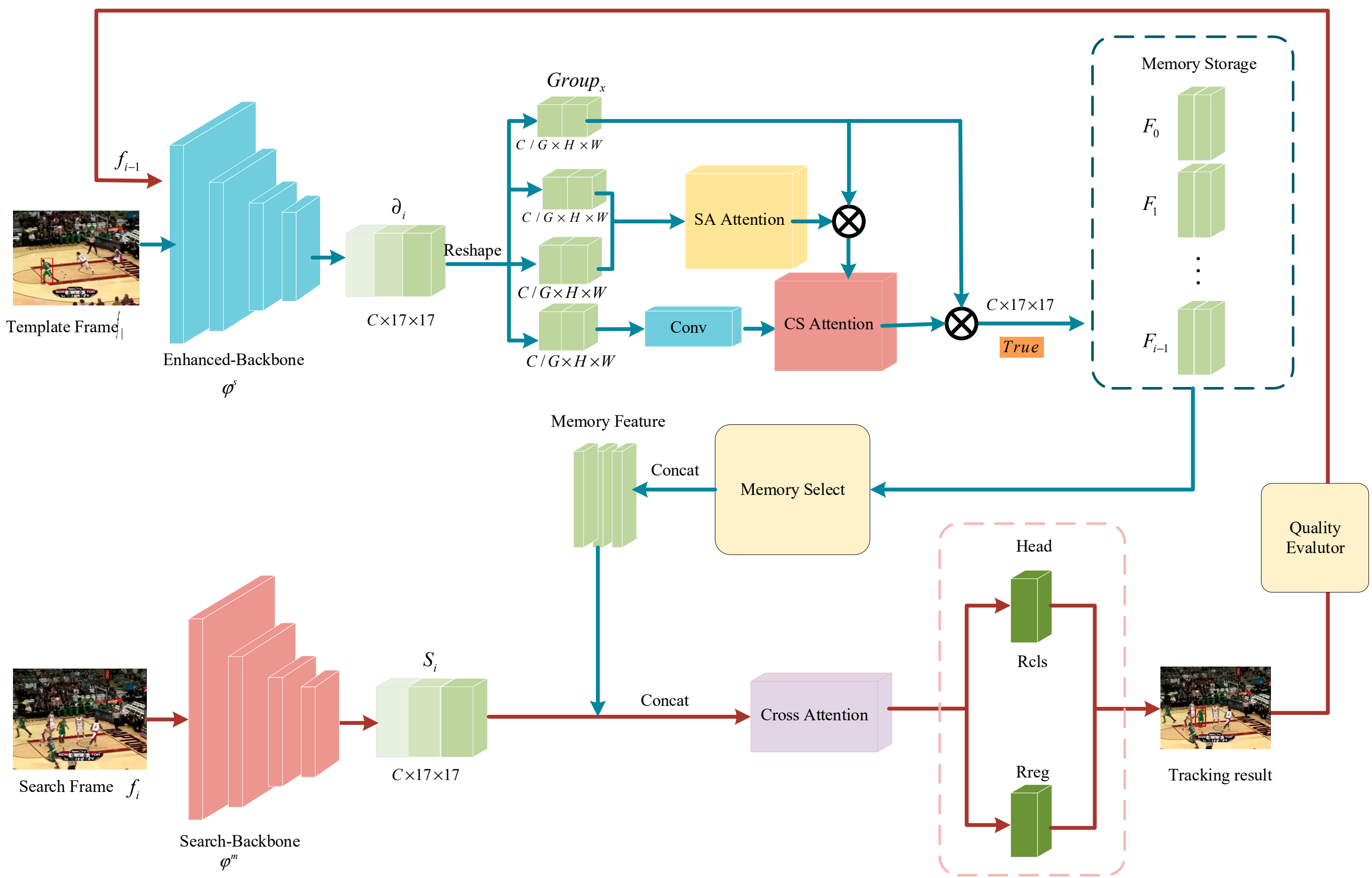

Figure 1. The architecture of our proposed method. Enhance the spatial location information of the feature, encode the global information in two spatial dimension directions, and capture the long-range interaction in different dimension directions. SA-Attention (spatial adaptive attention): SA Attention embeds accurate position information into the channel and captures long-range interactions in different dimensional directions. CS-Attention (cross-spatial attention): The fusion of context information at different scales enables the neural network to generate better pixel-level attention to advanced feature maps. Cross-Attention: The memory frame feature map is used as input to aggregate temporal and spatial context information. Cls (Classification): Classification branch, responsible for predicting the category of the target. Reg (Regression): Regression branch, responsible for predicting the location of the target.

For subsequent frames in the tracking sequence, the input of the feature enhancement branch is updated to the tracking result frame of the previous frame. Regardless of the quality assessment result of the tracking result frame, feature extraction and enhancement processing will be performed first. However, the enhanced features will be stored in the memory storage only when the quality judger evaluates the result as True (i.e., judged as high quality). At the same time, the search branch continues to receive new search frames as the input for real-time feature extraction and tracking processing.

The feature enhancement module enhances and mines the spatial features of the input to the feature enhancement branch through spatial adaptive attention and cross-spatial attention. Following this, it aggregates the enhanced multiple memory frame features using cross-attention. The aggregated feature map then contains both temporal and spatial context information of the target. On the other hand, the memory quality selection module is primarily composed of memory storage, a memory selection mechanism, and a memory quality evaluator.

3.2. Feature Extraction

Search Frame feature extraction. The algorithm uses GoogleNet [39] as the feature extraction network. The structure of the two feature extraction networks, φ^s and φ^m, are similar but not exactly the same, which belongs to the pseudo-twin network structure. The search branch takes the search frame f_i ("i" is the sequence number of the video frame) as the input and performs feature extraction through the feature extraction network φ^s. In order to facilitate the subsequent calculation, the extracted feature reduces the feature dimension to 512 through a nonlinear convolution layer (defined as h^s) and the final search frame feature map $S_i \in \mathbb{R}^{c\times h\times w}$ is obtained. The calculation process is shown in Formula (1).

$$S_i = h^s(\varphi^s(f_i)) \tag{1}$$

In Formula (1), s_i represents the search frame and h^s represents the nonlinear convolution layer.

Memory feature extraction. In the feature enhancement branch, the memory frame f_{i-1} is used as the input. In order to eliminate the interference of background information, we dynamically generate a mask $B^{f_{i-1}} \in \mathbb{R}^{H\times W}$ according to the tracking or real results (where the target area is defined by the bounding box, the pixels in the box are marked as 1, and the pixels outside the box are 0); then, we use a convolutional layer φ^n to embed the mask into $B^{f_{i-1}}$ in a specific space. At the same time, the input historical frame m^i is mapped to the same space as $B^{f_{i-1}}$ through the convolution layer φ^r. Then, the main elements of the two are added to suppress the background information of the historical frame. Finally, we obtain the feature map $\partial_i \in \mathbb{R}^{c\times h\times w}$ of the historical frame. In this paper, the background information marked as 0 is not directly removed because the tracking results are not necessarily accurate. If the area information marked as 0 is directly and completely removed by using $B^{f_{i-1}}$, the useful information will be lost. The same was as the search branch, the feature dimension ∂^i needs to be reduced to 512 through the nonlinear 1×1 convolution layer (expressed as h^s). The calculation process is shown in Formula (2).

$$\partial_i = h^s(\varphi^m(\varphi^n(B^{m_i}) + \varphi^r(f_{i-1}))) \tag{2}$$

"+" represents the sum of elements, φ^m represents the feature extraction network, and φ^r and φ^n represent the two-dimensional convolution operation.

3.3. Feature Enhancement Channel Attention

Traditional channel attention mechanisms primarily focus on the significance of individual channels, overlooking the interactions and correlations among them. This constraint poses a challenge to the model, which weakens the model's ability to capture detailed and complex features and fully understand global context information. Although some channel attention mechanisms perform well on feature representation, they model cross-channel relationships by dimensionality reduction (such as using 1×1 convolution), which reduces computational and memory consumption and helps to capture cross-channel interactions but may also lead to the loss of useful information, especially in tasks requiring fine feature recognition. To delve deeper into the spatial information of targets and enhance the feature representation ability and prediction accuracy of the model, we integrate spatial adaptive attention (SA Attention) with cross-space attention (CS Attention). This combination strategy enables the model to more fully understand and utilize the spatial characteristics of the target, thereby improving its performance.

Spatial adaptive attention (SA Attention). The specific channel dimension of the input feature map is transformed into a part of the batch dimension. This strategy not only ensures the integrity of each piece of channel information but also optimizes the computational efficiency and effectively reduces the processing cost. According to Figure 2, firstly, the input feature map $\partial_i \in \mathbb{R}^{c\times h\times w}$ is mapped to M sub-feature maps along the channel dimension (i.e., the depth dimension of the feature map) and its shape is re-adjusted to $Group_x \in \mathbb{R}^{C/M\times H\times W}$. Through the above dimension reconstruction, we can perform operations on each independent sub-feature map separately, which enables spatial semantic features to be evenly assigned to each feature group.

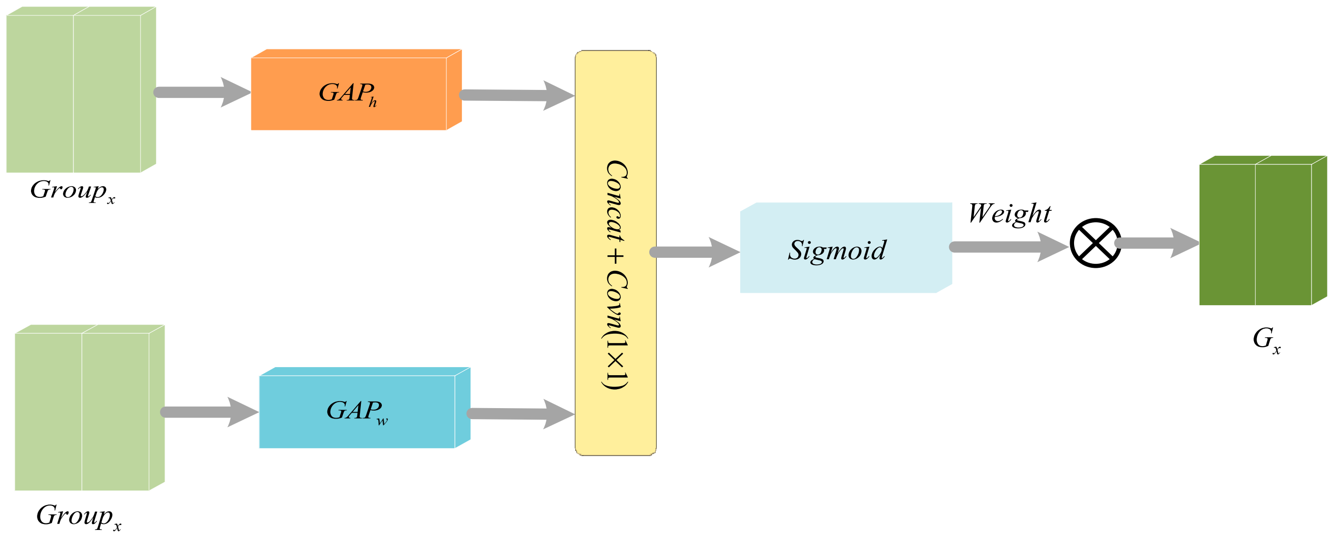

Figure 2. Structure diagram of spatial adaptive attention.

This approach essentially bypasses the common dimensionality reduction steps in traditional convolution operations, thereby maintaining richer feature information. Then, the $Group_x$ is averaged in the horizontal and vertical directions for each group, as shown in Formulas (3) and (4), and the feature maps x_h and x_w in both directions are obtained. The average distribution in the horizontal and vertical directions can be captured from the feature map, providing location-aware information. Then, the information in the horizontal and vertical directions is combined by the splicing operation, so that the model can consider the importance of different directions in the feature map at the same time. Then, the model can adaptively learn the complex relationship between features through the 1×1 convolution layer and fuse the attention features in height and width to obtain the fusion feature map $h_w \in \mathbb{R}^{(C/M)\times(H+W)\times 1}$. The process is shown in Formula (5).

$$x_h = GAP_h(Group_x) \quad (3)$$

$$x_w = GAP_w(Group_x) \quad (4)$$

$$h_w = \phi(x_h \oplus x_w) \quad (5)$$

"$\oplus$" represents the *Concat* operation. GAP_h represents the adaptive average pooling operation along the vertical direction. GAP_w represents the adaptive average pooling operation along the horizontal direction. φ represents the 1×1 convolution operation and then, the fused feature map is re-divided along the $(H + W)$ dimension and the channel attention maps $X_h \in \mathbb{R}^{(C/M)\times H\times 1}$ and $X_w \in \mathbb{R}^{(C/M)\times 1\times W}$ are obtained by feature transformation; through the feature transformation operation, the model can better understand and utilize the information in different directions in the feature map. Then, X_h and X_w are used to map the attention weights to the range of [0, 1] by using the sigmoid activation function and the attention weights in the horizontal and vertical directions are obtained, which indicates the importance of the corresponding position. Finally, the obtained weights are multiplied by the input feature map $Group_x$ and the features of each position are weighted and combined to achieve feature fusion and adjust the importance of features according to the attention allocation. The process is shown in Formula (6). The above operation embeds the accurate position information into the channel to capture the long-range interactions in different dimensional directions and finally obtains the weighted feature map $G_x \in \mathbb{R}^{C/M\times H\times W}$.

$$G_x = \mathrm{sigmod}(X_h) \times sigmod(X_w \cdot P) \times Group_x \quad (6)$$

"$\times$" represents the element multiplication operation, P represents a reshape operation, which reshapes the weights to the same dimension as the input $Group_x$ for weighting, and Sigmoid is an activation function.

Cross-space attention (CS Attention). SA attention ignores the importance of interaction between the entire spatial locations and the receptive field of 1×1 kernel convolution is limited. Therefore, in order to increase the receptive field, one of the group feature maps $Group_x$ is used to obtain a larger receptive field feature G_3 through a convolution operation with a convolution kernel size of 3×3,which can be used to capture local cross-channel interactions and expand the feature space.

As shown in Figure 3, the input of cross-spatial attention is the feature graph $G_x \in \mathbb{R}^{C/M \times H \times W}$ with location information obtained by SA attention and the feature graph $G_3 \in \mathbb{R}^{C/M \times H \times W}$ with a larger receptive field. Firstly, G_x is group-normalized to obtain g_x, which helps to improve the scale of features and facilitate subsequent calculations. The g_x and the global spatial information are encoded by the two-dimensional adaptive average pooling operation, respectively.

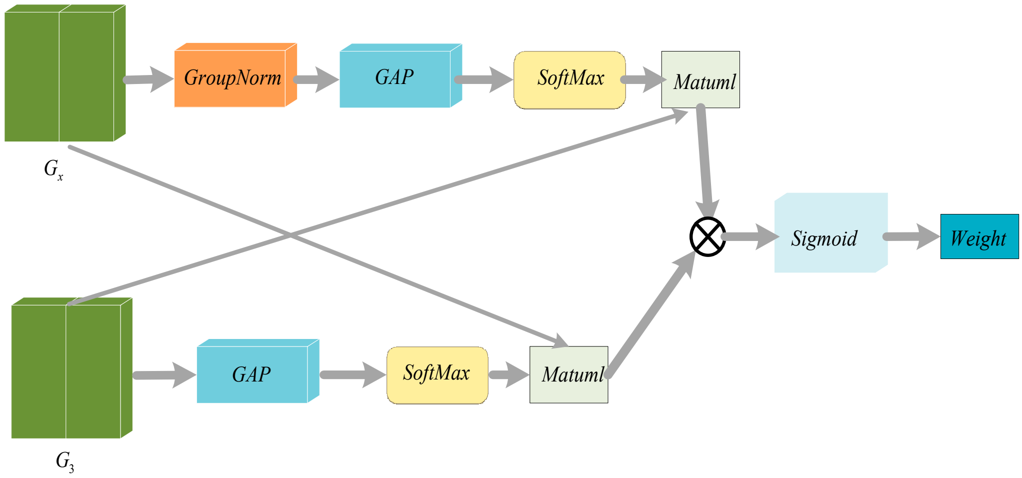

Figure 3. Structure diagram of cross-space attention.

In order to improve the computational efficiency, the natural nonlinear function Softmax of the two-dimensional Gaussian mapping is used to fit the above linear transformation at the output of the two-dimensional global average pooling to obtain the attention weights X_{11} and X_{21}. Among them, two branches perform shape conversion operations while performing pooling operations and convert them into corresponding dimensional shapes for subsequent calculations, as shown in Formulas (7) and (8).

$$X_{11} = S^m(GAP(gn(G_x))) \tag{7}$$

$$X_{21} = S^m(GAP(G_3)) \tag{8}$$

GAP is the average pooling, gn represents the group normalization, and S^m represents the softmax function. Remodeling g_x with G_3 obtains g_{x1}, G_{31}. The reshaping operation here is to obtain the same shape as the attention weight matrix X_{11} and X_{21} for subsequent matrix calculations. Then, the matrix multiplication between the two sets of attention weights X_{11} and X_{21} and the features G_{31} and g_{x1} is calculated, respectively, and the results are added to obtain the final weight. The first spatial attention map is obtained by multiplying X_{21} with g_{x1}, which collects spatial information of different scales at one stage and then the second spatial attention map is obtained by multiplying X_{11} with G_{31}, which retains the whole accurate spatial location information. Finally, the output feature maps in each group are added to generate a set of two spatial attention weight values and then the Sigmoid function is used to capture the pixel-level pairwise relationship to obtain the attention weight after feature fusion. This is then multiplied with the input, highlighting the global context of all pixels, and the final feature map $f_m \in \mathbb{R}^{C\times H\times W}$ is obtained. The calculation process is shown in Formulas (9) and (10).

$$Weight = sig((X_{11} \otimes g_{X1}) + (X_{21} \otimes G_{31})) \quad (9)$$

$$f_m = Group_x \times Weight \quad (10)$$

Among them, $\otimes$ represents the matmul matrix multiplication operation, sig represents the activation function Sigmod function, and "+" represents the matrix addition operation.

3.4. Cross-Attention Mechanism Based on Temporal Context

Cross-attention is a technique widely used in deep learning, especially when dealing with tasks with multiple input sequences or channels. It can establish associations between different input sequences or channels, capture long-distance dependencies, and improve the expression and generalization ability of the model. However, some transformer-based cross-attention only considers a single pixel or local information. Although it improves the tracking accuracy, it has low computational efficiency and is susceptible to noise interference, which affects the tracking stability. In order to enhance the tracker's resistance to scene changes, we take the historical frame (that is, memory frame) information into account and use the appearance information of the target in different time series to improve the tracker's adaptive ability.

As shown in Figure 4, the multiple memory frame feature maps selected by the memory selection mechanism are connected in series to obtain the key and value of the joint memory frame feature map after series, which are recorded as $K \in \mathbb{R}^{512\times(625\times t)}$ and $V \in \mathbb{R}^{512\times(625\times t)}$, respectively. The process is shown in Formula (11).

$$K = V = Concat(F_0, F_{b1}, \cdots, F_{i-1}) \quad (11)$$

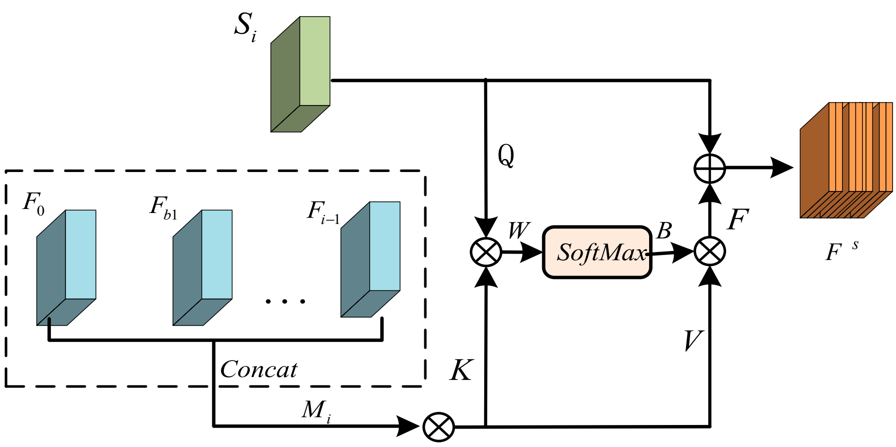

Figure 4. Structure diagram of the cross-attention mechanism based on temporal context.

Here, the *Concat* in the above formula represents a concatenation operation. Then, joint memory frame feature map and the search area feature is calculated to obtain the correlation between the two.

Firstly, the value $Q \in \mathbb{R}^{512\times 625}$ is obtained according to the feature S_i of the search area and then the similarity between the obtained key K of the joint memory feature map and the Q of the search area is calculated to obtain a similarity matrix $W \in \mathbb{R}^{625\times 625\times t}$. Then, W is input into the Softmax function. The original similarity score matrix is converted into a probability distribution matrix B, where each probability represents the similarity between the pixels of the search area and the concatenated features at that pixel. The calculation process is shown in Formula (12).

$$B = softmax(KQ^T) \quad (12)$$

Finally, the matrix B is multiplied by the V of the joint memory frame feature to obtain F and the key information required for tracking is extracted from the memory frame. Then, F is connected with the features S_i of the search area along the channel dimension to generate the final composite feature map $F^S \in \mathbb{R}^{512\times25\times25}$. The calculation process is shown in Formulas (13) and (14).

$$F = V \otimes B \tag{13}$$

$$F^s = F \oplus S_i \tag{14}$$

The memory frame information of the target contains the appearance information of the target in different time sequences, that is, the time context information. This contextual information is fused as the input of cross-attention and the required target information is extracted by calculating the similarity with the search frame, which improves the adaptive ability of the tracker. This cross-attention method based on time context has a certain robustness in target occlusion or changing scenes.

3.5. Memory Quality Selection Mechanism

3.5.1. Quality Evaluator

Online updates are crucial in memory networks to capture changes in targets and backgrounds. However, the introduction of noise templates will lead to model degradation and tracking drift. Therefore, it is necessary to determine which frame can be updated. Although the introduction of dynamic information improves adaptability, it may also bring noise. Some trackers judge the reliability of the tracking results by designing an evaluation mechanism, which is usually evaluated based on criteria such as confidence score or maximum response to determine whether to update it. However, these standards are risky. To this end, a quality evaluator is introduced to integrate multiple standards to output a score to determine whether the current frame is updated, as shown in Figure 5.

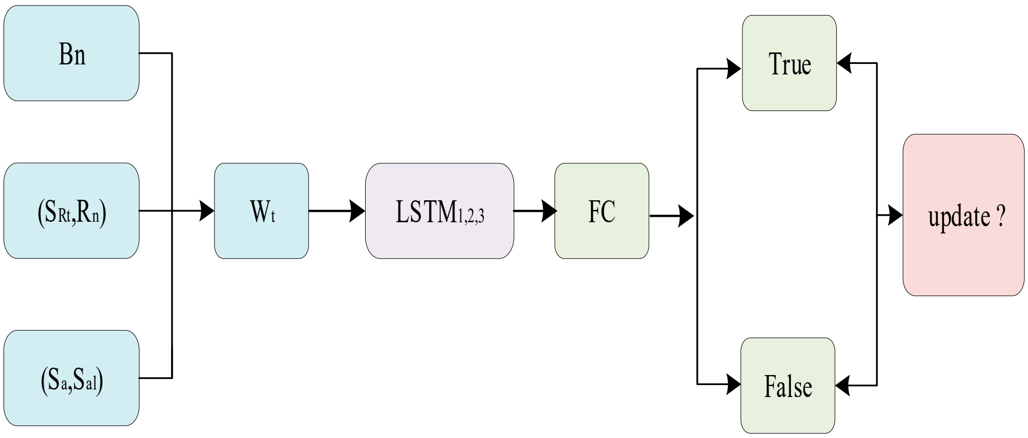

Figure 5. Structure diagram of the quality evaluator.

Before using the quality estimator, we first need to obtain the relevant information, which mainly includes the target bounding box B_n in the N frame, the response vector R_n, the tracking result frame S_r, and the initial template frame F^r. Among them, the target bounding box $B_n = [x_n, y_n, w_n, h_n]$, $[x, y]$ represents the coordinate information of the upper left corner and $[w, h]$ represents the length and width of the bounding box, respectively. The bounding box information of a single target provides the location and size information of the target and the bounding box of multiple consecutive frames can be used to determine the motion state of the target. For the response vector R_n, the confidence score S_{Rt} in object tracking is defined as the maximum value R of the response graph in the target bounding box, as shown in Formula (15).

$$S_{Rt} = Max(R) \tag{15}$$

In order to overcome the instability of the confidence score in the model prediction, the convolutional layer is used to dig deep into the response map of the input data to obtain the discriminant vector R_n, as shown in Formula (16).

$$R_n = Conv(R, W) \tag{16}$$

W represents the parameters of the convolutional layer. In order to improve the discriminative power of the Conv model, two appearance scores are defined to measure the difference between the tracking result and the initial frame positive sample. One of the scores is based on MetricNet to calculate the Euclidean distance S_a between the tracking result S_r and the template frame F^r, reflecting the appearance difference between the two, see Formula (17).

$$S_a = \|(M(S_r; W) - M(F_r, W)\|_2 \tag{17}$$

Among them, W is the CNN network parameter, M is the MetricNet network, and the second is a density-based score S_{al}, as shown in Formula (18). This score provides a stable similarity evaluation when the appearance of the target changes dramatically because the LOF-based calculation takes into account the reachable distance I of multiple positive samples, reflecting multiple aspects of the appearance of the target. Combining the two appearance scores S_a and S_{al} can obtain more stable appearance information.

$$S_{al} = LoF(M(S_r; M), M(I, W)) \tag{18}$$

Finally, the target result tracking box information Bn, discriminant vector (S_{Rt}, R_n) and appearance cues (S_a, S_{al}) are aggregated into a sequential matrix W_t,and the three-stage cascaded LSTM model is used to further mine the tracking state in this information. If it is True, it can be stored in memory for subsequent tracking. If it is False, it cannot be stored in memory.

Multilevel Long Short Term Memory Recurrent Neural Network (LSTM). By introducing gating mechanisms and memory cells, LSTM can selectively retain and update information and effectively capture long-term dependencies in sequences. Here, LSTM is used for cascading to deeply mine the timing information of the input matrix and enhance the model's ability to process sequence data.

After obtaining the sequence feature W_t in the previous section, it is input into the cascaded LSTM and the time step is gradually reduced to extract the sequence information more finely and focus on the nearest frame, so as to better capture the short-term dynamic changes, avoid paying too much attention to the historical information far away, and improve the response-ability and prediction accuracy of recent events. The processed result is input into two fully connected layers together with the appearance score to obtain a binary score, which is used to determine whether the current memory frame is suitable for updating.

3.5.2. Memory Selection Strategy

The memory selection strategy plays a central role in filtering and screening the key features of the target. Under the background noise and occlusion interference in complex scenes, the target features are often blurred and unstable. Therefore, efficient extraction and utilization of target information has become the key to tracking.

In the tracking task, for the current frame F_m, we fix the first frame F_1 (providing accurate target state) and the previous frame F_{m-1} (closest to the current frame, high feature similarity) and the remaining frames are selected from the memory pool according to the following strategy to assist tracking. First, the remaining candidate frames in the memory pool are divided into $M-2$ non-overlapping segments $F_{b1},\ F_{b2},\ F_{b3},\ \cdots,\ F_{b(m-2)}$ and the intermediate frame are selected as the key frame, as shown in Formula (19). This strategy not only integrates the change information of the target between consecutive frames and effectively avoids the risk of under-fitting but it also filters the noise and instability factors, greatly reduces the number of frames processed by the algorithm, and reduces the time cost. This method significantly improves the stability and accuracy of tracking and provides an efficient and reliable solution for object tracking in complex scenes, where $\Delta_j \in [0,1)$ and $j \in \{1,2,\cdots,M-2\}$. In our experiments, M is set to 7 and we simply set $\left\{\Delta_i = \frac{1}{2}|1 \leq i \leq M-2|\right\}$.

$$b_j = \left\lfloor \left\lfloor \frac{i-1}{M-2} \right\rfloor \times (j + \Delta j) \right\rfloor \tag{19}$$

3.6. Head Network

The traditional anchor point method is limited by the preset anchor point set and lacks flexibility. By removing this limitation, the anchor-free prediction head significantly improves the adaptability to the various sizes and shapes of the target. In this paper, it is introduced into the tracker to enhance the flexibility and accuracy of tracking. The classification branch distinguishes the target from the background by the features of the temporal cross-attention output F^s and the anchor-free regression branch directly predicts the target bounding box. The classification branch combines the classification response maps $Rcls \in \mathbb{R}^{1\times H\times W}$ and centrality response maps $Rctr \in \mathbb{R}^{1\times H\times W}$ to obtain the final classification response and the regression branch generates the regression response $Rreg \in \mathbb{R}^{4\times H\times W}$ to jointly improve the tracking performance. According to the predicted target position in the regression response graph, the result frame with the target result tracking box is generated.

4. Experiment

The experiment was conducted on a server equipped with 6 NVIDIA GeForce RTX 3090 graphics cards, a configuration that ensures high performance and computational efficiency for the experiment. The proposed tracker is evaluated on five benchmarks: OTB2015 [40], UAV123 [41], LaSOT [42], TrackingNet [43], and GOT-10k [44]. The key information of the dataset used in our experiment is listed in detail in Table 1.

Table 1. The dataset-related information used to test the performance of the tracker mainly includes the number of videos, the number of frames, and the number of challenge types in the dataset.

Dataset	Number of Videos	Number of Frames	Number of Challenge Types
OTB2015 [40]	100	59 k	11
UVA123 [41]	123	113 k	12
LaSOT [42]	1400	3.5 M	14
TrackingNet [43]	30,643	14.43 k	15
GOT-10k [44]	10,000	1.5 M	6

4.1. Construction of the Training Model Architecture

Training stage. The modules required for training the proposed network mainly include three main parts: backbone network, neck network, and head network. The following is the introduction to each module:

The backbone network is divided into two branches: feature enhancement and query, both based on fine-tuning InceptionV3. The core of the Neck network is AdjustLayer, which effectively reduces the amount of calculation and promotes feature fusion. The head network is responsible for classification and regression and generates the output required for tracking.

4.1.1. Training Details

Training stage. The training set of the FQ-track contains five datasets, which are ILSVRC VID [45], ILSVRC DET [45], LaSOT, GOT-10k, and COCO [46]. In order to ensure fair and reliable data, we solely utilized the GOT-10k dataset for model training and subsequently conducted evaluations of the trained model on the test set of GOT-10k. The tracker stochastic gradient descent (SGD) training network trains 20 Epochs. The whole training process takes about 25 h. The model adopts a phased learning rate adjustment strategy in the training phase. In the early stage of training, the learning rate first increased linearly from 0.01 to 0.08 to quickly adjust the model weight, subsequently, the learning rate was smoothly attenuated from 0.08 to 0.000001 in the subsequent 19 rounds to achieve fine-tuning of the model and improvement in the generalization ability.

4.1.2. Loss Function Correlation

In the training phase of the tracker, three loss functions are used, which are classification loss (FocalLoss), center point loss (SigmoidCrossEntropyCenterness), and regression loss (IOULoss), aiming to accurately guide the model to track the target stably and accurately in continuous video frames.

Classification Loss (FocalLoss): It alleviates sample imbalance as classification loss, enhancing model discrimination for target categories by dynamically weighting difficult vs. easy samples. The calculation process is shown in Formula (20).

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{20}$$

Here, p_t is the prediction probability of the model for category t, α_t is the weight of category t, and γ is the focusing parameter.

Center Point Loss (Sigmoid Cross-Entropy Centerness): Sigmoid cross-entropy loss guides the model to learn the target center position, ensuring accurate center point offset prediction. The calculation process is shown in Formula (21).

$$SCELossctr = \frac{\sum\limits_{i=1}^{B}\sum\limits_{j=1}^{HW} BCE_{\text{with\_logits}}(pred_{ij}, label_{ij})\cdot mask_{ij}}{\max\left(\sum\limits_{i=1}^{B}\sum\limits_{j=1}^{HW} mask_{ij}, 1\right)}\cdot weight \tag{21}$$

"B" is the batch size. "HW" is the height of the feature map multiplied by the width. $pred_{ij}$ is the predictive centrality logits on location (i, j). $Lable_{ij}$ is the true centrality label on position (i, j). $BCE_{with\_\log its}$ denotes the use of a binary cross entropy function with logits and $mask_{ij}$ is a binary mask used to ignore the loss of background or unlabeled regions, $weight$ is a hyperparameter used to weight the loss.

Regression loss (IOU Loss): The IoU between prediction and target boxes is optimized as the loss function for accurate bounding box regression. The calculation process is shown in Formulas (22) and (23), as follows:

$$IOU_{ij} = \frac{\text{int}er_{ij}}{union_{ij}} \tag{22}$$

$$IOULoss_{reg} = \frac{\sum_{i=1}^{B}\sum_{j=1}^{N}\left[-\log(IOU_{ij}+\varepsilon)\right]\cdot mask_{ij}}{\max(\sum_{i=1}^{B}\sum_{j=1}^{N} mask_{ij}, 1)}\cdot weight \tag{23}$$

Among them, $\text{int}er_{ij}$ is the intersection area of the jth prediction box and the target box in the ith sample and $union_{ij}$ is their union area. "B" is the batch size, "N" is the number of prediction boxes in each sample, IOU_{ij} is the IOU value of the jth prediction box and the target box in the ith sample, $mask_{ij}$ is a binary mask used to ignore the loss of background or unlabeled regions, and $weight$ is a hyperparameter used to weight the loss.

Finally, we add the above three loss functions together to obtain the overall loss function, which is used for backpropagation to optimize the model parameters. The calculation process is shown in Formula (24):

$$TotalLoss = FL(p_t) + SCELossctr + IOULoss_{reg} \tag{24}$$

4.2. Compare with a State-of-the-Art Tracker

OTB2015 [40]: As an important benchmark in the field of visual tracking, the OTB2015 dataset provides strong support for evaluating and improving tracking algorithms. OTB2015 provides a variety of evaluation indicators to measure the performance of tracking algorithms, including the Precision, Success Rate, and Average Overlap. The tracking algorithm designed in this chapter verifies the effectiveness of the designed algorithm by comparing it with nine advanced trackers on the OTB2015 dataset, including SiamAttn [9], MDNet [14], SiamRPN++ [18], SiamFC++ [19], SiamCAR [20], DaSiamRPN [23], SiamBAN [25], SiamGAT [33] and Ocean [47]. As shown in Figure 6, it can be seen that FQ-Track (the abbreviation of the tracker proposed in this chapter) achieves better performance than other classic trackers. Compared with SiamGAT, the success rate and accuracy were increased by 1.8% and 3%, respectively. Compared with DaSiamRPN, the success rate and accuracy were increased by 7% and 7.1%, respectively.

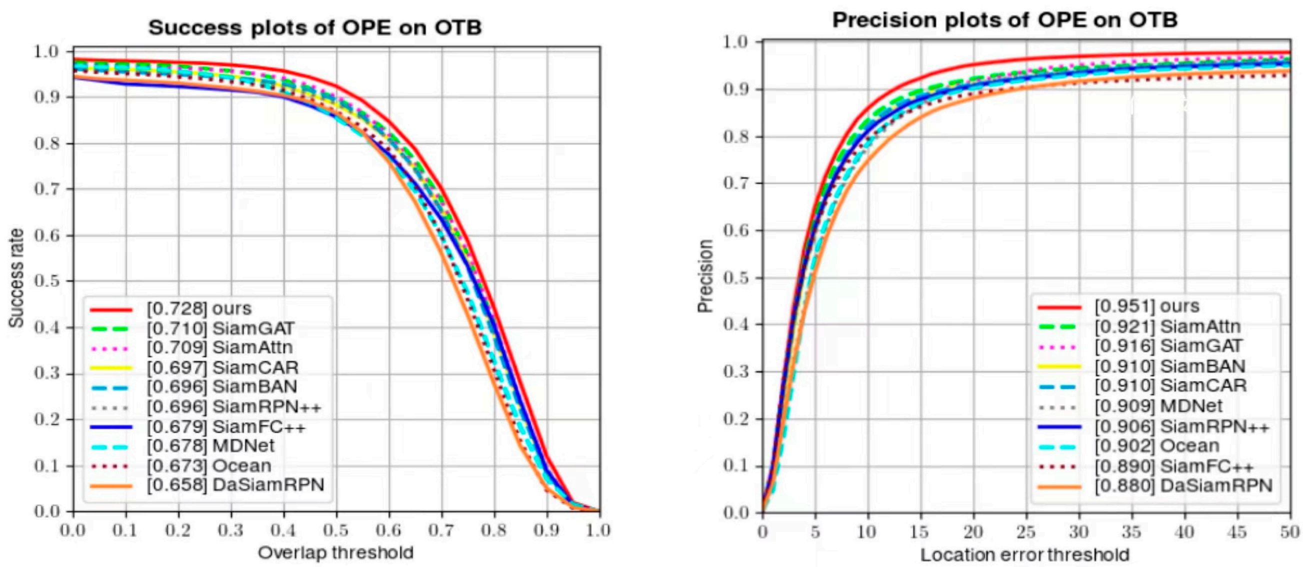

Figure 6. Plots show comparisons of the FQ-track with other advanced trackers on the OTB2015.

GOT-10k [44]: The GOT-10k dataset has the characteristics of being large-scale, highly diverse, and a fair evaluation. It has been widely used in the research and evaluation of target tracking. Many advanced tracking algorithms have been tested and evaluated on this dataset and achieved significant performance improvements. The tracking algorithm designed in this chapter is also tested and evaluated on the GOT-10k dataset and compared with other advanced tracking balls. As shown in Table 2, FQ-Track has a 0.7% and 7.2% improvement in the average overlap rate and success rate threshold of 0.75 compared with SiamGAT [33] and a 1.4% improvement in the average overlap rate compared with STEM [31]. Compared with DiMP [7], the average overlap rate, a success rate threshold of 0.50 and a success rate threshold of 0.75 were improved by 2.3%, 0.4%, and 6.8%, respectively.

Table 2. On the GOT-10k test benchmark, we conducted a performance evaluation and comparison of our proposed tracker against various state-of-the-art methods, focusing on the metrics of Average Overlap (AO) and Success Rate (SR) at thresholds of 0.5 and 0.75. The data values presented in the table reflect the performance of each method, with higher values indicating superior performance. To differentiate intuitively, the top three outstanding results are highlighted in red, blue, and green colors, respectively.

Tracker	AO (%)	SR (0.5, %)	SR (0.75, %)
ECO [48]	31.6	30.9	11.1
SiamFC [16]	34.8	35.3	9.8
SiamRPN [17]	36.7	42.5	10.3
SiamRPN++ [18]	51.7	61.6	32.5
SiamCAR [20]	56.9	67.0	41.5
SiamRAAN [49]	0.579	0.685	0.447
SiamFC++ [19]	59.5	69.5	47.9
Ocean [47]	61.1	71.1	47.3
DiMP [7]	61.1	71.7	49.2
STEM [31]	62.0	72.1	-
ATOM [6]	63.4	40.2	55.6
SiamGAT [33]	62.7	74.3	48.8
OURS	63.4	72.1	56.0

LaSOT [42]: Experiments on LaSOT dataset. The LaSOT dataset contains more than 1400 video sequences and the total number of frames exceeds 3.5 million. The LaSOT dataset mainly evaluates the performance of the tracker through success rate, accuracy, and normalization accuracy. As shown in Table 3, FQ-Track achieves 61.4%, 64.0%, and 69.1% in success rate, accuracy, and normalized accuracy, respectively. Compared with ten comparable trackers, our tracker has the best performance in success rate, accuracy, and normalized accuracy.

Table 3. On the LaSOT dataset, the proposed tracker has undergone a performance comparison with classic trackers introduced in recent years, focusing on metrics such as success rate (AUC). We have highlighted the top three outstanding results in red, blue, and green, respectively. In this comparison, the size of the data in the table directly reflects the actual effectiveness of each tracker, with higher values indicating superior performance.

Tracker	Suc. (%)	Prec. (%)	Norm. Prec. (%)
MDNet [14]	39.7	37.3	46.0
SiamADT [50]	50.6	51.1	-
SiamRPN++ [18]	49.5	49.3	57.0
GlobaTrack [51]	51.7	52.8	59.7
SiamBAN [25]	51.4	52.1	59.8
ATOM [6]	51.4	50.5	57.7
SiamCAR [20]	51.6	52.4	61.0
Ocean [47]	52.6	52.6	61.0
SiamGAT [33]	53.9	53.0	63.3
SiamSRT [52]	55.6	57.3	64.8
OURS	61.3	63.9	69.1

UVA123 [41]: The UAV123 dataset contains 123 carefully selected video sequences taken by drones under different environmental conditions, so they capture a variety of natural and man-made scenes observed from the drone's perspective. The particularity of this perspective makes the dataset full of challenges, such as fast motion, perspective change, occlusion, target loss, and small target tracking. In this chapter, the proposed tracker is compared with seven advanced trackers. The results are shown in Table 4. It can be seen that the FQ-Track tracker achieves a success (AUC) score of 0.64, which is still significantly better than the recent competitors SiamBAN [25], SiamCAR [20], SiamRPN++ [18], and ECO [48] running at real-time speed.

Table 4. On the UAV123 test set, we compare the proposed tracker with a series of relatively advanced trackers in terms of success rate (AUC). To clearly demonstrate the advantages, we highlighted the top three outstanding results in red, blue, and green, respectively. The larger the numerical value in this table, the more exceptional the tracking performance.

ECO [48]	Spiking SiamFC++ [53]	SiamRPN++ [18]	SiamCAR [20]	SiamBAN [25]	ATOM [6]	DiMP [7]	OURS
52.5	0.578	61.2	61.4	63.1	64.3	65.4	64.0

TrackingNet [43]: TrackingNet is a large dataset designed to evaluate single-target tracking algorithms. It contains more than 30,000 video sequences, each of which is labeled with a bounding box of the tracking target. These video sequences cover a wide range of scenes and conditions, including indoor and outdoor environments, different lighting conditions, motion patterns, target size changes, occlusion, and other challenges. Therefore, TrackingNet provides a very challenging test platform for tracking algorithms, which can comprehensively evaluate the performance of the algorithm in real scenarios. The proposed tracker is compared with other state-of-the-art trackers on the TrackingNet dataset and the results are shown in Table 5. The tracking results of our tracker improved by 0.9%, 3.3%, and 1.5%, respectively, in terms of success rate, normalization accuracy, and accuracy compared with TrDiMP.

Table 5. On the TrackingNet test benchmark, we conducted a performance evaluation of the proposed tracker against various outstanding methods, focusing on metrics such as success rate. To visually demonstrate the advantages, we highlighted the top three outstanding results using red, blue, and green colors. The data in the table directly reflect the performance of each method, with higher values indicating better results.

Tracker	Suc. (%)	Prec. (%)	Norm. Prec. (%)
SiamFC [16]	57.1	53.3	66.3
ATOM [6]	70.3	64.8	77.1
D3S [54]	72.8	66.4	76.8
SiamRPN++ [18]	73.3	69.4	80.0
DiMP [7]	74.0	68.7	80.1
KYS [55]	74.0	68.8	80.0
STEM [31]	74.6	70.7	81.6
SiamFC++ [19]	75.4	70.5	80.0
PrDiMP [38]	75.8	70.4	81.6
AutoMatch [56]	76.0	72.6	-
TrDiMP [57]	78.4	73.1	83.3
OURS	79.3	76.4	84.8

4.3. Ablation Studies

The influence of the channel attention mechanism based on feature enhancement: In order to verify the effectiveness of the channel attention mechanism based on feature enhancement, this chapter compares the tracker performance with and without the mechanism on the OTB2015 and LaSOT datasets. The results are shown in Table 6.

Table 6. Comparison results of trackers with and without the feature-enhanced channel attention mechanism on OTB2015 and LaSOT. W represents the use of the memory quality selection mechanism and W/O represents no use. The larger the data value in the table, the better the effect.

Dataset	W/0 Feature-Enhance	W Feature-Enhanced
OTB2015 success (%)	71.7	72.4
OTB2015 precision (%)	91.4	94.0
LaSOT success (%)	60.4	61.0
LaSOT precision (%)	63.0	63.7

Compared with the tracker without a feature enhancement channel attention mechanism, the success rate and accuracy of the tracker using this mechanism are increased by 0.7% and 2.6% on OTB2015 and 0.6% and 0.7% on LaSOT, which effectively proves the effectiveness of the feature enhancement strategy.

The influence of the memory quality selection mechanism on the performance of the tracker: In order to verify the effectiveness of the quality evaluators and the memory selection mechanism proposed in this chapter, this chapter verifies the performance of the tracker using the memory quality selection mechanism and the tracker without the memory quality selection mechanism in the OTB2015 dataset. The comparison results are shown in Table 7.

Table 7. Comparison results of trackers with and without the memory quality selection mechanism on OTB2015, W represents the use of the memory quality selection mechanism, and W/O represents no use.

Dataset	W/0 Memory Quality Selection	W Memory Quality Selection
OTB2015 success (%)	72.4	72.8
OTB2015 precision (%)	94.0	95.1

Compared with the tracker without a memory quality selection mechanism, the tracker with this mechanism not only improves the success rate by 0.4% but also improves the accuracy by 1.1%. Through ablation experiments, we clearly see this improvement and fully demonstrate the effectiveness of the new method.

The influence of the number of memory frames on the tracking results: The selection of the number of memory frames in the object tracking algorithm based on the feature-enhanced spatial-temporal memory network is critical to the tracking results. In order to verify the influence of the number of memory frames on the performance of the tracker, a set of experiments was performed on OTB2015, as shown in Table 8.

Table 8. Comparison of trackers with different numbers of context features on OTB2015.

Dataset/Number of Frames	3	4	5	6	7	8
OTB2015 success (%)	71.4	71.6	71.6	72.0	**72.4**	70.9
OTB2015 precision (%)	91.6	92.4	93.2	93.8	**94.2**	92.5

The experimental results clearly reveal the specific law of the tracker performance changing with the number of memory frames: the initial performance increases with the increase in the number of frames and reaches the optimal at 7 frames and then the performance decreases with the increase in the number of frames, which verifies the key influence of the number of memory frames on the tracking effect.

After in-depth analysis, the following conclusions can be drawn when the number of memory frames is small, the feature information contained is not enough to support the tracker for effective object tracking, resulting in poor performance. When the number of memory frames is too large, too many memory frames may contain noisy data, which will interfere with the normal operation of the tracker and reduce its performance. Therefore, it can be inferred that the number of memory frames has an important influence on the performance of the tracker. Too few or too many memory frames may adversely affect the performance of the tracker. In order to obtain the best tracking effect, it is necessary to find a balance point in the number of memory frames that can ensure the richness of feature information and avoid the interference of noise.

4.4. Visual Analysis of Tracking Process

Visual results analysis: The visualization of our proposed algorithm is shown in Figure 7. The representative video sequence frames in the OTB2015 dataset are selected to prove the effectiveness of the algorithm. In the Basketball video sequence, that is, the first line in the figure, when the target's motion appearance changes, it will be disturbed by the background information, resulting in inaccurate tracking. Compared with other trackers, ours tracker shows better performance. The green box in the figure is the tracking result bounding box of the tracker in this chapter and the red box is the real value bounding box. In the Bird1 video sequence, when the target suddenly disappears or similar objects appear, other trackers are prone to drift or tracking failure but our tracker can still track accurately. In the Coke video sequence, when the target moves, our tracker can track the target more accurately and continuously than the Ocean and SiamCAR trackers. In the Human3 video sequence, our tracker can continue to accurately track the real target box compared to other trackers in the background clutter or sudden disappearance scene. In summary, it can be concluded from the diagram that our tracker shows competitive performance with advanced trackers in the case of appearance deformation, disappearance, and similar object interference caused by target motion.

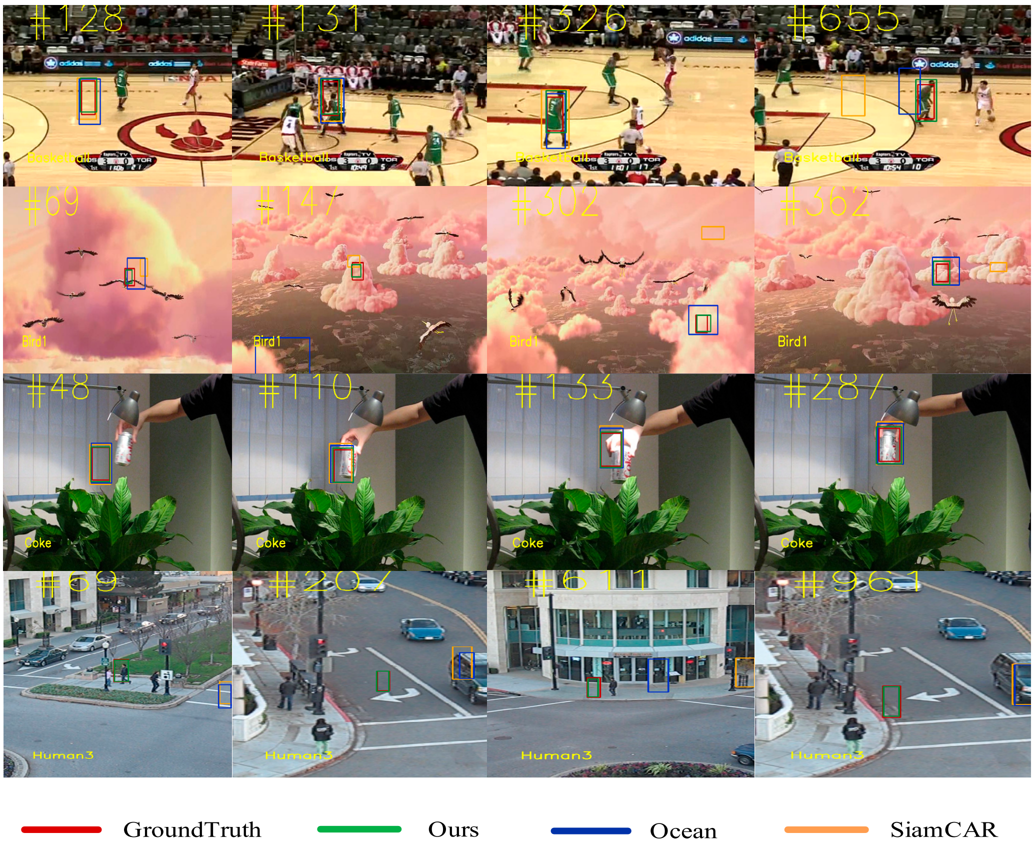

Figure 7. This method is compared with the visualization results of other advanced trackers on OTB2015.

5. Conclusions

In this paper, a novel tracking framework is proposed. By introducing spatial adaptive attention and cross-space attention mechanism, the spatial representation of the target is effectively enhanced and cross-attention is used to aggregate temporal and spatial context information to improve the adaptability of the tracker to the appearance deformation of the target. At the same time, the proposed memory quality selection mechanism can reduce the influence of noise, enrich the diversity of target features, and improve the adaptability and discrimination ability of the model. Experiments on several challenging benchmark datasets show that this method achieves comparable performance to advanced trackers.

Although the method proposed in this paper improves the performance and robustness of the tracker to a certain extent when dealing with complex scenes such as fast motion and occlusion of the target, these improvements have not completely conquered all challenges. Specifically, with the improvement in tracking accuracy, the computational complexity of the algorithm increases accordingly, resulting in a certain limitation in tracking speed. At present, it can only reach the processing rate of 35 frames per second (fps), which may constitute a certain bottleneck in the application scenarios with high real-time requirements.

It is particularly, it is worth noting that in extremely complex scenes, such as extremely cluttered background, unusual target motion speed, or experiencing large deformation, the tracker may still face the risk of tracking failure. This is mainly because although our model has achieved initial results in improving the richness and accuracy of feature extraction, these advances are still incremental improvements and have not yet reached a fundamental breakthrough. As the key cornerstone of tracking performance, the accuracy and richness of features still have a broad space for further improvement, which is also an important direction to be overcome in our future research work.

Looking forward, we will focus on refining attention mechanisms, especially integrating multi-modal data (vision, sound, and text) with a transformer to enhance tracking adaptability. This multi-dimensional environment offers detailed context for trackers, boosting their perception and understanding. By leveraging the self-attention in transformer, we will improve tracking accuracy and robustness in complex scenarios, advancing future tracking technologies.

Author Contributions: Conceptualization, H.Z. and M.Z.; methodology, M.Z.; validation, J.Z., Z.C. and M.Z.; data curation, Z.C. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62072416), the Key Research and Development Special Project of Henan Province (221111210500), and the Key Technologies R&D Program of Henan Province (232102211053, 242102211071, 242102210142).

Data Availability Statement: Data are contained within the article.

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

References

1. Lu, X.; Ma, C.; Ni, B.; Yang, X. Adaptive region proposal with channel regularization for robust object tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 1268–1282. [CrossRef]
2. Chen, G.; Pan, G.; Zhou, Y.; Kang, W.; Hou, J.; Deng, F. Cor-relation filter tracking via distractor-aware learning and multi-anchor detection. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 4810–4822. [CrossRef]
3. Xie, F.; Wang, C.; Wang, G.; Cao, Y.; Yang, W.; Zeng, W. Correlation-Aware Deep Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 8741–8750.
4. Kadam, P.; Fang, G.; Zou, J.J. Object Tracking Using Computer Vision: A Review. *Computers* **2024**, *13*, 136. [CrossRef]
5. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning Dynamic Siamese Network for Visual Object Tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1781–1789.
6. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In Proceedings of the CVPR, Long Beach, CA, USA, 16–20 June 2019; pp. 4655–4664.

7. Bhat, G.; Danelljan, M.; Van Gool, L.; Timofte, R. Learning discriminative model prediction for tracking. In Proceedings of the ICCV, Seoul, South Korea, 27 October–2 November 2019; pp. 6181–6190.
8. Li, P.; Chen, B.; Ouyang, W.; Wang, D.; Yang, X.; Lu, H. Gradnet: Gradient-guided network for visual object tracking. In Proceedings of the ICCV, Seoul, South Korea, 27 October–2 November 2019; pp. 6162–6171.
9. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable siamese attention networks for visual object tracking. In Proceedings of the CVPR, Seattle, WA, USA, 14–19 June 2020; pp. 6728–6737.
10. Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; Ling, H. SwinTrack: A simple and strong baseline for transformer tracking. In Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22), Red Hook, NY, USA, 28 November–9 December 2021; pp. 16743–16754.
11. Zhang, L.; Gonzalez-Garcia, A.; van de Weijer, J.; Danelljan, M.; Khan, F.S. Learning the model update for siamese trackers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4010–4019.
12. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning spatiotemporal transformer for visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 10448–10457.
13. 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 Visionand Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 13608–13618.
14. Nam, H.; Han, B. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
15. Xie, F.; Yang, W.; Zhang, K. Learning Spatio-Appearance Memory Network for High-Performance Visual Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
16. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016; pp. 850–865.
17. Li, B.; Yan, J.; Wu, W. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Ecognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
18. Li, B.; Wu, W.; Wang, Q. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4282–4291.
19. Xu, Y.; Wang, Z.; Li, Z. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12549–12556.
20. Guo, D.; Wang, J.; Cui, Y. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6269–6277.
21. 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, 16–20 June 2019; pp. 4591–4600.
22. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H.S. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the CVPR, Long Beach, CA, USA, 16–20 June 2019; pp. 1328–1338.
23. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018; pp. 101–117.
24. Fan, H.; Ling, H. Siamese cascaded region proposal networks for real-time visual tracking. In Proceedings of the CVPR, Long Beach, CA, USA, 16–20 June 2019; pp. 7952–7961.
25. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese box adaptive network for visual tracking. In Proceedings of the CVPR, Seattle, WA, USA, 14–19 June 2020; pp. 6668–6677.
26. Liu, F.; Perez, J. Gated end-to-end memory networks. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Long Papers, Valencia, Spain, 3–7 April 2017; Volume 1, pp. 1–10.
27. Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; Weston, J. Key-value memory networks for directly reading documents. *arXiv* **2016**, arXiv:1606.03126.
28. Yang, T.; Chan, A.B. Learning dynamic memory networks for object tracking. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018; pp. 152–167.
29. Choi, J.; Kwon, J.; Lee, K.M. Deep meta learning for real-time target-aware visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 911–920.
30. Zhang, H.; Chen, J.; Nie, G. Light regression memory and multi-perspective object special proposals for abrupt motion tracking. *Knowl.-Based Syst.* **2021**, *226*, 107127. [CrossRef]
31. Zhou, Z.; Li, X. Object Tracking via Spatial-Temporal Memory Network. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 2976–2989. [CrossRef]
32. Lukezic, A.; Vojir, T.; Zajc, L.C. Discriminative correlation filter tracker with channel and spatial reliability. *Int. J. Comput. Vis.* **2018**, *126*, 671–688. [CrossRef]
33. Guo, D.; Shao, Y.; Cui, Y. Graph attention tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 9543–9552.

34. Wang, Q.; Teng, Z.; Xing, J. Learning attentions: Residual attentional siamese network for high performance online visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4854–4863.
35. Wang, Q.; Zhang, M.; Xing, J. Do not lose the details: Reinforced representation learning for high performance visual tracking. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018.
36. Subakan, C.; Ravanelli, M. Attention Is All You Need In Speech Separation. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 21–25.
37. Chen, X.; Yan, B.; Zhu, J. Transformer tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 8126–8135.
38. Danelljan, M.; Gool, L.V.; Timofte, R. Probabilistic regression for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 7183–7192.
39. Szegedy, C.; Liu, W.; Jia, Y. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
40. Wu, Y.; Lim, J.; Yang, M.-H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
41. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer International Publishing: Cham, Switzerland, 2016; pp. 445–461.
42. Fan, H.; Lin, L.; Yang, F. 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, 16–20 June 2019; pp. 5374–5383.
43. Muller, M.; Bibi, A.; Giancola, S. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 300–317.
44. 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]
45. Russakovsky, O.; Deng, J.; Su, H. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
46. Lin, T.Y.; Maire, M.; Belongie, S. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13. Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
47. Zhang, Z.; Peng, H.; Fu, J. Ocean: Object-aware anchor-free tracking. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXI 16. Springer International Publishing: Cham, Switzerland, 2020; pp. 771–787.
48. Danelljan, M.; Bhat, G.; Shahbaz Khan, F. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
49. Xin, Z.; Yu, J.; He, X. SiamRAAN: Siamese Residual Attentional Aggregation Network for Visual Object Tracking. *Neural Process. Lett.* **2024**, *56*, 98. [CrossRef]
50. Wang, F.; Cao, P.; Wang, X. SiamADT: Siamese Attention and Deformable Features Fusion Network for Visual Object Tracking. *Neural Process. Lett.* **2023**, *55*, 7933–7950. [CrossRef]
51. Huang, L.; Zhao, X.; Huang, K. Globaltrack: A simple and strong baseline for long-term tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11037–11044.
52. Huang, B. Searching Region-Free and Template-Free Siamese Network for Tracking Drones in TIR Videos. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15. [CrossRef]
53. Xiang, S.; Zhang, T.; Jiang, S. Spiking SiamFC++: Deep spiking neural network for object tracking. *Nonlinear Dyn.* **2024**, *112*, 8417–8429. [CrossRef]
54. Lukezic, A.; Matas, J.; Kristan, M. D3s-a discriminative single shot segmentation tracker. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 7133–7142.
55. Bhat, G.; Danelljan, M.; Van Gool, L. Know your surroundings: Exploiting scene information for object tracking. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXIII 16. Springer International Publishing: Cham, Switzerland, 2020; pp. 205–221.
56. Zhang, Z.; Liu, Y.; Wang, X. Learn to match: Automatic matching network design for visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 11–17 October 2021; pp. 13339–13348.
57. Wang, N. Transformer Meets Tracker: Exploiting Temporal Context for Robust Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 1571–1580.