Visual target tracking based on fractional – order bidirectional hybrid attentional feature fusion

Yao Fu[1]**, Yilu Wang**[2]
[1]School of Information Science and Engineering, Shenyang Ligong University, Shenyang, China
[2]Department of Marine Science and Technology, Northwestern Polytechnical University, Xi An, China
[1]Corresponding author
E-mail: [1]*fuyao2662@163.com*, [2]*yiluwang6588699@163.com*

Received 17 October 2024; accepted 20 November 2024; published online 22 January 2025
DOI https://doi.org/10.21595/jme.2024.24620

Check for updates

Abstract. This paper mainly designs and implements target tracking based on fractional – order feature fusion to solve tracking drift and tracking box jumping in complex scenes. Firstly, the starting point and overall structure of the model design were introduced. Secondly, in response to the low information utilization of the feature extraction network in the target tracking framework at the local scale of the target, a fractional – order node function based attention CNN hybrid attention extraction module was proposed to improve the robustness of target tracking. Finally, overall performance was quantitatively and qualitatively evaluated on multiple evaluation datasets with various advanced trackers. The results showed that the algorithm proposed in this paper had a high tracking advantage under severe changes in scale morphology, dynamic blurring, and similar interference attributes.

Keywords: visual target tracking, fractional-order, feature fusion, bidirectional network.

1. Introduction

The target tracking task has been continuously pushed into the arena of deep learning development in recent years due to the landing of applications such as intelligent public security, unmanned driving, and area tracking. Target tracking is a perceptual basic task, and improving its recognition performance is a necessary way to study the subsequent major problems in the cognitive field. Over the long time, the target tracking field has primarily addressed four major challenges: the drastic changes in the morphological and state characteristics of target objects, the interference from natural conditions such as lighting, the accurate discrimination and stable tracking of similar target identities, and the inherent noise of sensors. Each of these challenges has spurred targeted research efforts. However, achieving a high-precision and efficient tracker remains a challenging and formidable problem.

In the past decade, the design of trackers predominantly relied on dual-stream structures, from plain similarity matching with shared weight ground to feature fusion of deeply related features. The emphasis in target tracking has consistently been on the feature response of both the template and the retrieved image. In previous work, the more classical neural network-based feature fusion for siamese models still relies on the cross-correlation operation, which computes the feature similarity between the template and the retrieved region to obtain the target region with the maximum response [1]. However, this cross-correlation operation is based on the invariant assumption, and the connectivity mechanism of convolutional networks is locally residual, which imposes limitations on the modeling scale and hinders the comprehensive consideration of global features. Along with the rapid rise of attention-based detection means, the field of target tracking has been influenced by the work related to information processing in this kind of cerebral computation, and multiple attention mechanisms are also introduced into feature enhancement and fusion, so as to improve the model's power of characterization and discrimination of the object of interest. Following the introduction of encoder-decoder-based transformer networks, a subset of research has focused on combining convolutional networks with transformers. By utilizing the encoded features of the transformer, these approaches aim to enhance the precision and breadth of information learning, empowering the target tracking models with robust feature representation capabilities, which in turn influences subsequent predictions. DeepMind proposes to add spatial attention to the convolutional network structure to add spatial attention branch to adaptively learn the data [2]. CBAM proposes dual hybrid attention mechanism of channel and spatial, inheriting SENet to adaptively deal with the degree of contribution of feature channels. Jiang Yingjie and others replaced the conventional backbone of the Siamese network with the swin transformer, fully utilizing the advantages of the encoder-decoder structure for the fusion of dual-stream features, thereby making the tracker more robust against occlusion interference [3]. In response to the deficiency of global feature attention in the Siamese family of trackers, it is proposed that leverages the contextual dependency relationships learned by the self-attention modules within the transformer encoder to capture rich global features from both input branches [4]. To address the robustness limitations of convolutional networks due to the difficulty in establishing time-domain correlations specific to the target tracking task, it is proposed to utilize the transformer encoder to fuse spatiotemporal domain information, thereby obtaining a joint spatiotemporal representation [5]. It stands out from a crowd of work that utilizes the transformer encoder to improve and process convolutional network features for tracking, as it innovatively proposes to completely eliminate convolutional kernels [6]. It focuses on the issue of information loss of correlation operations in target tracking tasks and proposes a tracker with a Siamese convolutional network as the backbone and a feature fusion based on multiple attention mechanisms, offering insights for enhancing the cumulative error reduction [7].

In this background, this paper proposes a bidirectional tracker based on fractional-order hybrid attention feature fusion. With the fractional order differential theory and the attention property of the transformer encoder, multi-scale feature representation and information fusion are obtained to achieve target tracking to cope with the robustness of complex scenes with similar occlusion and unstable lighting conditions. Specifically, this paper implements the following: designing an attention fusion encoder, FoBAF-Encoder; to enhance the interest feature extraction, globally fusing the attention distributions under different channels, while introducing fractional order differentiation to handle the attention network node function to improve the adequacy of the convolutional block for usable information extraction in the region of weakly-gradient texture features. Finally, the cross-entropy loss function to optimize the network model is used for iterative training to complete the target tracking task.

2. Materials and methods

The framework of this paper encompasses three main components: the hybrid fractional-order attention feature extraction module (HFAF), the fractional-order bidirectional attention fusion encoder (FoBAF), and the tracking prediction head. The framework chooses a dual-stream Siamese structure, sharing the template branch and search branch of weight parameters in parallel. Firstly, the simplified and modified residual network ResNet50 is used as the backbone of feature extraction, inherited from Siamese's system, so that the features of both branches are processed by fractional-order feature extraction fusion network, which makes the features extracted and learned at this stage more reasonably relevant and discriminative, and avoids model degradation of deep network while extracting the features of template and tracking object. Finally, the tracking is realized through the classification regression network. The structure of the overall framework is shown in the following Fig. 1.

2.1. HFAF hybrid feature extraction module

For different dataset inputs, the pixel sizes to be processed by the target tracking model varies. Therefore, it is initially necessary to undergo operations for the preprocessing module of the model, such as crop and reshape to unify the input sizes of the bidirectional network. Specifically, in the following Fig. 2, it is hypothesized that the input image of the search branch is x_S, and the input of the template branch is x_T. After the crop operation, the reasonably cropped target-size images $X_S$and X_T are obtained. At this time, the tracking object of our concern is in the center between X_S and X_T.

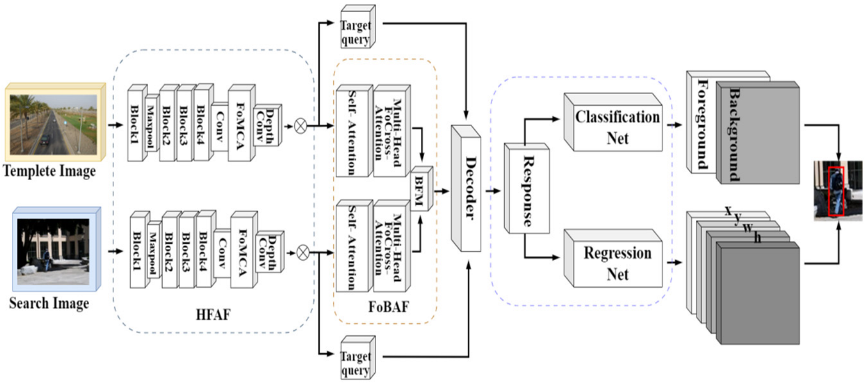


Fig. 1. FoBAF-T frame structure

In numerous classical target tracking tasks, convolutional networks are commonly employed as the primary architecture for feature extraction. With the prevalence of transformer networks, a retrospective examination of past research reveals that convolutional neural networks have the advantage of local modelling and spatial preservation on data with spatial structure. When serving as the backbone for feature extraction, CNN exhibits a light-weight nature that is currently lacking in Transformer models, in terms of parameter quantity and computational efficiency. Moreover, for images rich in spatial data structures, CNN naturally possesses advantages in hierarchical abstracted representations, handling local scales, and fine-grained textures. However, fully attention-based Transformer architectures offer superior advantages in the association modeling of sequential data and in handling long-range and historical dependencies, which are tasks with high dynamic requirements such as target tracking and virtual reality, and these benefits are difficult to match by convolutional neural networks. Therefore, in this paper, in light of the aforementioned research background and by integrating the merits of both in terms of different feature expressions, the attention mechanism is combined with the convolutional networks to obtain the feature extraction module – HFAF in the FoBAF-T framework.

The HFAF includes a Resnet50 feature learning phase and a feature aggregation phase. In the first phase, 1×1 convolution is used to map the features to a deeper space achieving parameter sharing for the purpose of increasing the network's nonlinear expressive ability. This also serves to reduce the number of parameters, thereby mitigating the risk of overfitting. In the second phase, fractional-order channel attention designed is introduced to capture the features of different channels in a fine-grained way. This allows the network to have greater flexibility in modeling complex distribution scenarios. Upon obtaining the channel attention weights, the hybrid convolutional pathway employs these weights to perform weighted fusion across the various channels, thereby achieving a fusion of channel attention with convolutional operations. Finally, to control the integration of the information from the two pathways, two learnable parameter controllers are added to nonlinearly integrate the information from the convolutional kernel attention pathway. The altered ResNet50 hybrid fractional-order channel attention structure adopted by the HFAF is shown in the following Fig. 2.

The first four stages of the original ResNet50 are retained and the down sampling step size is adjusted to 1 in stage4 to ensure the originality of the feature details. To ensure the richness and globality of feature extraction, the original convolutional layers are replaced by atrous convolution with a step size of 2 in this stage to expand the feeling field of the convolution operation. Let the residual block in this residual structure consist of two convolutional layers and a jump connection, and the output feature maps are y_T, y_S:

$$y_S = F(X_S, \{W_i\}) + X_S, \tag{1}$$

$$h_{y_S} = \frac{h_{X_S}}{8}, \tag{2}$$

$$w_{y_S} = \frac{w_{X_S}}{8}, \tag{3}$$

$$z_{y_S} = 1024, \tag{4}$$

where i is the number of layers, W_i is the weight parameter of the i-th convolution operation, F is the nonlinear mapping function constituted by two convolutional layers, z_{y_S}, w_{y_S}, h_{X_S}, w_{X_S}, z_{y_S} are the pixel sizes of the input and output images, respectively, and the target image is the same. The features X_{T_1}, y_T and X_{S_1}, y_S obtained from stage 3 and stage 4 are jointly fed into the encoder to achieve the extraction of features at different levels in order to improve the representation of the model for multi-scale features.

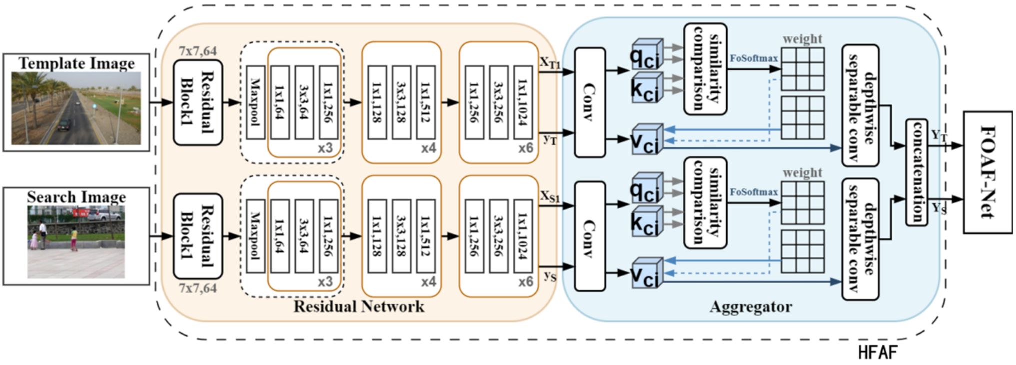


Fig. 2. HFAF structure diagram

2.2. FoMCA, fractional order multi-channel attention module

FoMCA belonging to HFAF is shown in Fig. 1. Let the input to the search branch of the fractional-order multichannel attention feature aggregation phase be Y_S, the mapping size of feature Y_S be $w \times h$, and the number of channels be z. The channel attention weights ω are:

$$\omega = \text{Softmax}[g(Y_S)] = \text{Softmax}\left[B\delta\left(B\left(W_1\left(g(Y_S)\right)\right)\right)\right], \tag{5}$$

$$g(Y_S) = \frac{\sum_{m=1}^{h}\sum_{n=1}^{w} Y_{S_{[:,m,n]}}}{w \times h} \in R^Z, \tag{6}$$

where Softmax is the activation function, $g(Y_S)$ is the global feature context, B denotes batch normalisation, W_2 is the ascending layer, W_1 is the descending layer, and δ denotes the ReLU operation. Due to the limitation of channel attention on multi-scale feature space, this paper aggregates multi-scale contexts on the basis of channel attention, and uses point-by-point convolution (PWConv) as a local channel attention aggregator to improve the characterisation ability of channel attention (henceforth referred to as CA) for small objects and locally weak signals. The local context $L(Y_S)$ is computed by the following equation:

$$L(Y_S) = B\left[PWConv_2\left(\delta\left(B\left(PWConv_1(Y_S)\right)\right)\right)\right]. \tag{7}$$

Next, the node function needs to be utilised to play a role between the similarity calculation and weighted combination to convert the similarity into attention weights. The softmax function is usually chosen to convert the similarity into a probability distribution so that the sum of all the attention weights is 1. The softmax function helps to ensure that the attention weights are normalised [6-7] so that they correctly represent the importance of the different input elements, which effectively captures the associations between the input elements and produces the appropriate attention weights.

Thus, the multichannel attention feature $Y_S{}'$ can be obtained as:

$$Y_S{}' = Y_S \otimes M(Y_S) = Y_S \otimes \mathrm{Softmax}\big(L(Y_S) \oplus g(Y_S)\big), \tag{8}$$

where $M(Y_S)$ denotes the attention weights generated after Y_S passes through the multi-channel attention network, which are weighted by the local context and global context.

The node function is processed within the improved multichannel attention module using fractional order differentiation. Caputo fractional order derivatives are used here to describe the nonlocal relationships during feature fusion. fractional order derivatives produce a memory effect in the attention module by capturing historical information. This memory effect allows the model to better adapt to changes in target appearance and noise interference in the scene. Specifically, the fractional order derivative retains more historical feature information, thus maintaining stable tracking performance in complex scenes such as target occlusion and illumination changes. In addition, the fractional order derivative can also enhance the robustness of the model, making the model more sensitive to small changes in the input data, thus improving the overall functionality of the system, where $\mathrm{Softmax} = p(Y_S) = \frac{e^{Y_S}}{\sum_{j=1}^{r} e^{Y_S}}$, is obtained by a first order derivation of $D^{0.5}p(Y_S) = \frac{1}{\Gamma(0.5)}\int_0^{Y_S}\frac{p^{(1)}(t)dt}{(Y_S-t)^{0.}}$:

$$p'(Y_S) = p(Y_S)\big(1 - p(Y_S)\big). \tag{9}$$

A 0.5 order derivative for $p(Y_S)$ is obtained:

$$D^{0.5}p(Y_S) = \frac{1}{\Gamma(0.5)}\int_0^{Y_S}\frac{p^{(1)}(t)dt}{(Y_S - t)^{0.5}} = \frac{1}{\Gamma(0.5)}\int_0^{Y_S} p(Y_S)\,(1 - p(Y_S))(Y_S - t)^{-0.5}dt. \tag{10}$$

Substituting $p(Y_S)$ and expanding by derivation yields:

$$\begin{aligned} D^{0.5}p(Y_S) &= \frac{1}{\Gamma(0.5)}\int_0^{Y_S}\frac{e^{Y_S}}{\sum_{j=1}^{r} e^{Y_S}}\left(1 - \frac{e^{Y_S}}{\sum_{j=1}^{r} e^{Y_S}}\right)(Y_S - t)^{-0.5}dt \\ &= \frac{1}{\sqrt{\pi}}\int_0^{Y_S}\frac{e^{Y_S}\left(\sum_{j=1}^{r} e^{Y_S} - e^{Y_S}\right)}{\sum_{j=1}^{r} e^{Y_S}{}^2}(Y_S - t)^{-0.5}dt = \frac{Y_S{}^{0.5} e^{Y_S}\left(e^{Y_S} - \sum_{j=1}^{r} e^{Y_S}\right)}{(\sum_{j=1}^{r} e^{Y_S})^2}. \end{aligned} \tag{11}$$

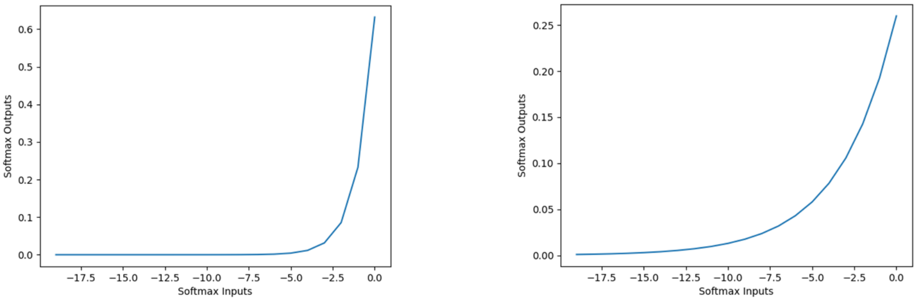
Similarly, the 0.3 and 0.8 order derivatives of $p(Y_S)$ are obtained, respectively:

$$D^{0.3}p(Y_S) = \frac{1}{\Gamma(0.3)}\int_0^{Y_S}\frac{p^{(1)}(t)dt}{(Y_S - t)^{0.3}}, \tag{12}$$

$$D^{0.8}p(Y_S) = \frac{1}{\Gamma(0.8)}\int_0^{Y_S}\frac{p^{(1)}(t)dt}{(Y_S - t)^{0.8}}. \tag{13}$$

The first order differential image of the nodal function Softmax and its 0.3 order differential image are shown below.

The Softmax function provides the probability distribution calculation for the output of the feature extraction network. Subsequently, multi-class decisions are implemented based on the Softmax probability calculation results, eventually generating the prediction outcome.



a) First order differential image

b) 0.3 order differential image

Fig. 3. Softmax function image

As can be seen from the above figure, after replacing the integer-order differentiation with fractional-order differentiation, the convergence of the curve tends to become gentle, which is due to the characteristics of fractional-order differentiation, which no longer singly considers the current value in the process of convergence, but builds on the historical information of the signal computation, describing some nonlinear and complex features that are not available in the function under the integer-order differentiation, and thus complex pixel feature probabilities is able to be considered from multiple perspectives and more stably.

From Fig. 3, it can be seen that the trend of change in σ based on fractional differentiation is different from that under integer order. Softmax(0.3) is less variable than Softmax(1). This indicates that the Softmax in fractional order has a stable and delicate convergence behavior, but its computational cost is not high. That is to say, the greater the rate of change of node functions within a unit is, the faster the model converges work, and under the same conditions with equal parameters, the shorter time it takes, but the probability calculation is incomplete and unstable, and it is not suitable for data bases with complex light texture conditions. The FoMCA module proposed in this paper is shown in Fig. 4.

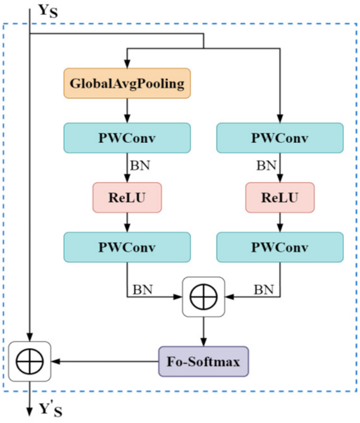
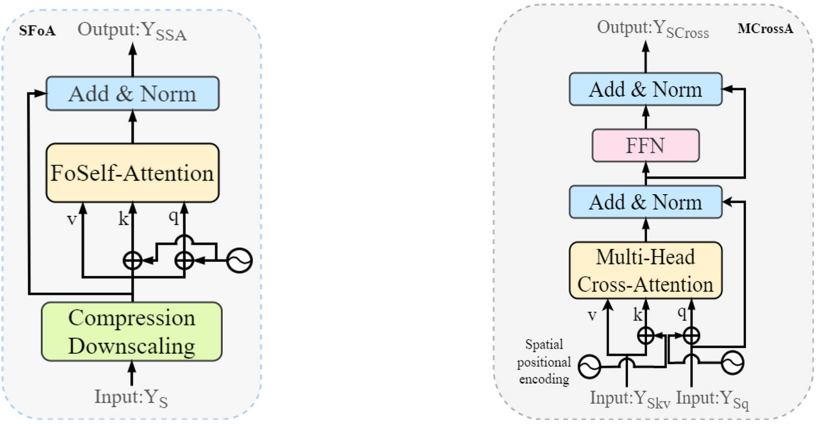


Fig. 4. HFAF structure diagram

2.3. FoBAF, fractional order attention feature fusion encoder

Upon obtaining the feature maps Y_S and Y_T from the HFAF module, these are fed into the fractional-order bidirectional attention fusion network (FoBAF-Net) to yield a bidirectionally fused enhanced feature map Y_F. Specifically, this paper designs a fusion network based on fractional-order attention to integrate features extracted from the template map and the search region. This network consists of three modules: the fractional-order self-attention module (FoSAM), the multi-head cross attention module (MCrossA), and the bidirectional fusion model (BFM), serving to preserve and restore texture detail features while focusing on the distribution of feature attention, respectively [8-10].



a) FoSAM b) MCrossA

Fig. 5. FoMSC multi-head attention mechanism

The above figures illustrate the main structure of the fractional-order bidirectional attention fusion network. Fig. 5(a) demonstrates the conversion of the multi-head self-attention from a conventional Transformer back to a self-attention mechanism, employing convolutional compression mapping to reduce the spatial dimensions of feature embedding vectors. This approach decreases the model's computational cost, accelerates the fusion process, and enhances the feature vectors processed through HAFA. Irrelevant semantic features derived from HAFA are discarded, while those with high relevance are preserved. Fig. 5(b) extends the fractional-order differential-based cross-attention module to multiple heads, capturing the correlations among input features and enhancing the model's capacity to represent the features of tracked targets in complex scenes [11-13].

2.3.1. FoSAM, fractional order compressed self-attention module

The dual-branch vectors Y_T, Y_S transmitted from the HAFA module, alongside the feature vectors X_{T1} and X_{S1} output from the third stage of ResNet50, exhibit distinct attributes in their feature information. After concluding the extraction process in the preceding three stages, the resultant features are relatively shallow, comprising a greater proportion of low-level textures, edge details, and color information, thus possessing a local advantage. In contrast, the processing in the fourth stage yields deeper, abstract high-level semantic information, characterized by complex global features. Following the fractional-order multi-head channel attention described in Section 2.1, the establishment of relationships between channels enhances the features of interest while suppressing irrelevant channels, thereby mitigating noise interference within challenging attributes. This paper aims to fuse features with the aforementioned distinct information to enhance the model's representational capability [14-17].

The self-attention mechanism can weight the importance of different features by calculating the similarity between them. The self-attention mechanism based on fractional-order node functions allows for nonlinear modeling of the similarity between features, providing a more flexible method for weight allocation. This enables the model to focus more on key target features, enhancing its performance capabilities. For the incoming variables Y_T, Y_S and X_{T1}, X_{S1}, along with the spatial dimensions h_{T1}, h_T, h_{S1}, h_S, w_{T1}, w_T, w_{S1}, w_S and channel counts d_{T1}, d_T, d_{S1}, d_S, a dimensionality reduction mapping is performed prior to entering the FoBAF fusion module to accelerate operation and reduce computational costs. Feature representations are typically high-dimensional and encompass rich information from the input data. Such data pose significant computational demands on the self-attention mechanism, particularly concerning image feature computations, where high complexity can lead to inefficiencies, contradicting the application requirements for target tracking. To better leverage these feature representations, embedding and encoding are employed to yield more meaningful feature representations, thus enhancing model performance. Consequently, through convolutional operations, the image features extracted by the HAFA module undergo spatial dimensional compression to obtain more compact vectors that are meaningful for subsequent attention encoding. Initially, dimensionality reduction mapping is conducted to ensure uniformity across $w \times h \times d$, followed by flattening to derive the corresponding $q_{X_{T_1}}$, $k_{X_{T_1}}$, $v_{X_{T_1}}$, q_{Y_T}, k_{Y_T}, v_{Y_T}, $q_{X_{S_1}}$, $k_{S_{T_1}}$, $v_{S_{T_1}}$, q_{Y_S}, k_{Y_S}, v_{Y_S}. Finally, the resulting embedded vectors are concatenated to yield the processed q_T, k_T, v_T, q_S, k_S, v_S:

$$\begin{aligned} q_T &= Concat\left(q_{X_{T_1}}, q_{X_T}\right), & q_S &= Concat(q_{X_{S_1}}, q_{X_S}), \\ k_T &= Concat\left(k_{X_{T_1}}, k_{X_T}\right), & k_S &= Concat(k_{X_{S_1}}, k_{X_S}), \\ v_T &= Concat\left(v_{X_{T_1}}, v_{X_T}\right), & v_S &= Concat(v_{X_{S_1}}, v_{X_S}). \end{aligned} \tag{14}$$

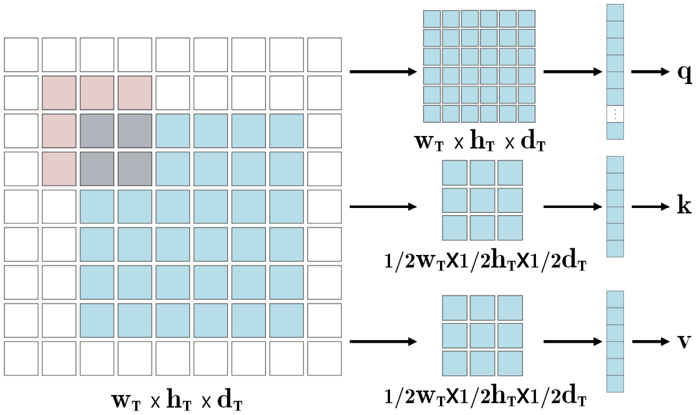

q
$\mathbf{w_T} \times \mathbf{h_T} \times \mathbf{d_T}$
k
$1/2\mathbf{w_T}\mathrm{X}1/2\mathbf{h_T}\mathrm{X}1/2\mathbf{d_T}$
v
$\mathbf{w_T} \times \mathbf{h_T} \times \mathbf{d_T}$
$1/2\mathbf{w_T}\mathrm{X}1/2\mathbf{h_T}\mathrm{X}1/2\mathbf{d_T}$

Fig. 6. Reduced dimensional mapping operation (light red parts indicate sliding windows)

From the implementation of FoMCA discussed in Section 2.2, it can derive the expression based on the fractional-order node function:

$$p^{(\alpha)}(Y_i) = \frac{1}{\Gamma_{(\alpha)}} \int_0^{Y_i} \frac{p^{(1)}(t)dt}{(Y_i - t)^{\alpha}}, \tag{15}$$

where Y_i represents the input to the fractional-order multi-head cross-attention module and i indicates the dual-stream branch $i \in \{T, S\}$. By reverting the multi-head self-attention in the Transformer model to single-head and replacing the node function within the self-attention mechanism, it can derive the self-attention weights:

$$SelfAttention(q, k, v) = FoSoftmax\left(\frac{q \times k^T}{\sqrt{d_k}}\right)v. \tag{16}$$

2.3.2. MCrossA, multi-head cross attention module

Expanding the cross-attention to multiple leads to its computational equation:

$$MutiHead(q,k,v) = Concat(head_1, head_2, \ldots, head_m)W^O. \tag{17}$$

Each $head_{\mathrm{m}}$ respectively:

$$head_m = FoCrossAttention\left(qW_m^q, kW_m^k, vW_m^v\right), \tag{18}$$

where (qW_m^q, kW_m^k, vW_m^v) is the weight matrix of $head_m$ and W^O is the spliced output weight matrix. The attention score matrix can be obtained from the following equation:

$$C = p^{(\alpha)}\left(\frac{qW^q(kW^k)^T}{\sqrt{d_k}}\right). \tag{19}$$

By weighting the query weight matrix with the attention contribution matrix C, it can achieve:

$$head_m = C \cdot vW^v. \tag{20}$$

Substituting into Eq. (18) yields the concatenated weight computation, resulting in the final representation of fractional-order multi-head cross attention.

2.3.3. BFM, bidirectional fusion module

After processing the features from both branches of the Siamese structure through the FoBAF-Attention module, it obtains the feature tensor Y_T from the template branch and the feature tensor Y_S from the search branch. These, along with the low-level feature vectors output from the Block3 stage of the HFAF module, are simultaneously input into the Bidirectional Fusion Module (BFM), ultimately resulting in the feature Y_F after multi-attention fusion.

The FoSAM and MCrossA modules are capable of globally capturing the relationships among various positions within the sequences of feature vectors received from HFAF. This contrasts with the fixed-size convolutional kernels used in ResNet50 at each layer. As a result, the feature tensors obtained after processing through the encoder's FoSAM and MCrossA possess a larger receptive field. During its learning process, FoSAM simultaneously considers the global information of the input sequence, thereby capturing contextual information over a broader range and expanding the receptive field of the feature tensor. In contrast, the receptive field of the feature vectors output from the third layer of ResNet50 is relatively small, primarily derived from local convolution operations constrained by kernel size and stride.

Herein, it is hypothesized that Y_T represents a feature map with a larger receptive field. More specifically, Y_T is a feature map, which belongs to the high-level semantic feature map, and X_{T1} is the learned residuals in ResNet50 Block3, which belongs to the low-level feature map. Based on the multi-scale attention channel weights M, this two-stage BAF feature iterative fusion strategy can be expressed as:

$$Y_F = Y_S \uplus Y_T = M(Y_S + Y_T) \otimes Y_S + \left(1 - M(Y_S + Y_T)\right) \otimes Y_T, \tag{21}$$

where $\uplus$ is the feature integration calculation. The integration weight $M(Y_S + Y_T)$ consists of real numbers between 0 and 1, allowing the network to perform soft selection or weighted averaging between Y_S and Y_T.

This paper summarizes different forms of feature fusion within deep networks in Table 1. G represents the global attention mechanism. Although various methods for feature fusion in different scenarios may differ in implementation, these details diminish once abstracted into mathematical forms. Thus, a carefully designed approach can unify these feature fusion scenarios, improving the performance of all networks by replacing original fusion operations with this unified method. Table 1 further illustrates those linear methods, such as addition and concatenation, do not rely on contextual information. Feature refinement and modulation are non-linear but can only partially perceive the input feature maps. In most cases, they primarily utilize high-level feature mappings. Unlike partially context-aware methods, fully context-aware methods encounter an unavoidable challenge: how to preliminarily integrate input features. As input to the attention module, the initial fusion quality significantly influences the final fusion weights. Since this remains a feature fusion issue, an intuitive approach is to employ another attention module to merge the input features.

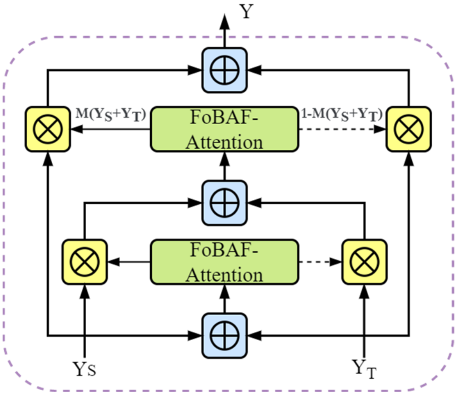


Fig. 7. Schematic diagram of BFM bidirectional feature fusion structure

After obtaining the fused feature vector Y_F, it is fed into the decoder for decoding to obtain the output map of similar responses. In this paper, classification regression is chosen to implement the tracker prediction head network.

Table 1. Example of complex scene properties

Context-aware	Type	Formulation	Example
None	Addition	$X + Y$	RestNet
	Concatenation	$W_A X_{:,i,j} + W_B Y_{:,i,j}$	U-Net
Partially	Refinement	$X + G(Y) \otimes Y$	SENet
	Modulation	$G(Y) \otimes X + Y$	GAU
	Soft Selection	$G(X) \otimes X + (1 - G(X)) \otimes Y$	Highway Networks
Fully	Modulation	$G(X, Y) \otimes X + Y$	SA
	Soft Selection	$G(X + Y) \otimes X + (1 - G(X + Y)) \otimes Y$	SKNet
		$M(X + Y) \otimes X + (1 - M(X + Y)) \otimes Y$	Ours

3. Multi-task learning network

In previous research, most trackers have trained classification and regression networks separately after feature extraction and fusion to predict target information in tracking images. This approach, which decomposes the tracking problem into multiple sub-tasks, has influenced numerous works and has indeed achieved advanced performance in some cases.

However, the independent forward propagation processes of feature extraction, fusion, and learning classification-regression modules lead to poor information interaction in the overall model. Under the same computational power and training mode, this modular design tends to cause the models performance and efficiency to hit a bottleneck. Therefore, a multi-task joint framework that simultaneously realizes feature extraction, fusion, and model training is necessary.

This paper employs a multi-task tracking prediction network with multi-output joint training. The tasks of target classification and information regression are integrated into one multi-task network. Specifically, a CNN with two output layers is used, where one output layer is for foreground-background classification, and the other is for normalized coordinate regression. The multi-task tracking prediction network consists of a shared HAFA section, a FoBAF section, and two output branches, each with a three-layer MLP (Multi-Layer Perceptron) equipped with hidden layers and ReLU activation functions.

During training, a dataset containing normalized coordinates and foreground-background labels can be used for joint training, achieving both classification and regression tasks simultaneously. To accomplish this, this paper adds the cross-entropy loss function and the regression loss function together as the total loss function of the multi-task network. Optimization is performed via backpropagation to minimize the total loss function and update the models' weights and biases. Compared to the method of independent training, the multi-task prediction network better leverages information interaction between the various modules of the model and reduces computational complexity and parameter count through shared mechanisms.

3.1. Training loss functions

Loss functions are essential in training neural networks. By measuring the difference between the predicted values and the true labels, loss functions capture the complex relationships between samples and labels. The loss function value reflects the error between the models' predictions and the ground truth, where a smaller value indicates more accurate prediction. Therefore, the goal is to reduce the loss value, turning it into an optimization problem to find the optimal solution, thus enhancing the models' predictive capabilities.

Since different loss functions apply to different tasks, they guide model learning in varying ways. In single-object visual tracking tasks, loss functions are typically divided into two categories: classification loss functions and regression loss functions. Cross-entropy loss is a classification loss function, while IOU loss (Intersection Over Union) and Smooth L1 loss are both regression loss functions.

3.1.1. Cross-entropy loss function

Borrowing the concept of "entropy" from thermodynamics, the cross-entropy loss function transforms the model into an entropy value to compare differences between model predictions. In classification problems, positive and negative samples in the sample labels are represented by 1 and 0, respectively. If the label value at a specific class position is 1, the sample belongs to that class, and vice versa. For binary classification and multi-class classification tasks, there are two forms of cross-entropy loss functions: Binary Cross Entropy (BCE) and multi-class cross-entropy loss. To calculate the models' loss, the model's output, which is a vector of class probabilities, is normalized to the range of 0 to 1, with each value representing the probability of a sample belonging to a class. In binary classification, the Sigmoid function is used for normalization, while in multi-class classification, the Softmax function is applied. Below are the two forms of the cross-entropy loss function.

In this paper, binary cross entropy (BCE) is used to train the tracking network to predict the probability that a given area belongs to the target object, as shown in following equation:

$$loss(BCE) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \ln(p_i) + (1 - y_i)\ln(1 - p_i)], \tag{22}$$

where, m denotes the number of samples; y_i denotes the true label of sample i; p_i denotes the predicted value of sample i. The multi-class cross-entropy loss function is expressed as in Eq. (23):

$$loss(CE) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n} y_{ij} \ln(p_{ij}), \tag{23}$$

where n denotes the number of classes, y_{ij} equals 1 if sample i belongs to class j, and 0 otherwise; p_{ij} represents the probability that sample i belongs to class j. Essentially, the multi-class cross-entropy loss is an extension of the binary cross-entropy loss across all classes.

3.1.2. GIOU loss function

Intersection Over Union (IOU) loss measures the error between the predicted bounding box and the ground truth bounding box, as shown in Fig. 8.

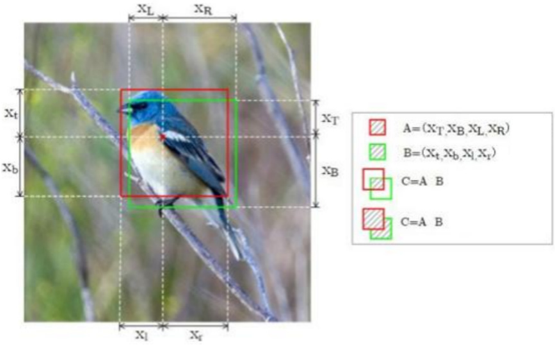


Fig. 8. IOU diagram

Eq. (24) illustrates the calculation process of the IOU loss function:

$$Loss_{IOU} = 1 - IOU(A, B) = 1 - \frac{C}{D}. \tag{24}$$

The parameters in Fig. 8 will not be elaborated here. However, the IOU loss function has some shortcomings: IOU cannot reflect the degree of overlap between the predicted and ground truth boxes when they do not intersect. In such cases, the loss value is calculated as 0, which does not produce a corresponding gradient, thereby failing to fulfill the purpose of learning and training. The GIOU loss function used in this section addresses the gradient issue when the two boxes do not overlap by introducing an additional penalty term ψ, as shown in Eq. (25):

$$Loss_{GIOU} = Loss_{IOU} + \psi = 1 + \frac{C}{D} + \frac{|C - D|}{|C|}. \tag{25}$$

Among them, $\psi = |C - D|/|C|$ is the penalty term, along with C and D, shares the same meanings as in Eq. (24). To some extent, LossGIOU mitigates and resolves the limitations of the IOU loss function.

3.2. Online update and window penalty

This paper makes minimal modifications to the tracking prediction head, inheriting the online template update and window penalty from previous work. In the inference phase, a conventional method is used by applying a Hanning window penalty to select the bounding box with the highest core as the regression result, further improving the models' accuracy and robustness.

4. Results

The algorithms in this paper are trained with OTB100, TrackingNet, and LaSOT. For instance, TrackingNet is trained with video frame image pairs 100 frames in the sequence TRAIN_0 to TRAIN_3. The implementation environment is Python 3.6, based on the Pytorch 1.5.1 framework, and the experiments are performed on a server with two RTX3060TiGPUs, and 64GB of memory. In this paper, a hybrid network is employed as the mainstay of the siamese structure. During the training process, the parameters pre-trained on ImageNet are utilized for initialization. Among them, the fusion encoder has a multi-head attention hidden layer with the dimension $d = 256$ and nHead = 8, and the hidden layer space dimension of the feedforward network is 2048. Dropout is set to 0.1 to reduce overfitting. The fully convolutional network in the bounding prediction consists of a five-layer convolutional block, and the decoder the bounding prediction uses a three-layer MLP with a hidden layer spatial dimension of 256. The training data consists of the training portion of the OTB100, LaSOT, and TrackingNet datasets. The target template image and the search region image are imported into the backbone network by the preprocessing module crop into the size of 128×128 and 320×320, respectively. Strategies such as pan-flip and brightness jittering are used for data enhancement. The overall training process is 1000 epochs using the AdamW optimizer. The initial learning rate of the overall network is 1E-5, the learning rate of other parameters is 1E-4, and the weight decay is 1E-4. The Batch size is set to 16, and training is conducted for 1,000 epochs. To guarantee stable training and convergence, strategies such as gradient clipping and learning rate decay are adopted. The learning rate is reduced to one-tenth of its original value after 400 epochs. The backbone network is fine-tuned with a learning rate that is one-tenth of that of the other parts.

4.1. Quantitative analysis

To evaluate the effectiveness of the algorithms in this paper, tests and evaluations are performed on the test sets of OTB100, LaSOT and TrackingNet, respectively.

Table 2. Example of complex scene properties

Sequence	Frame number	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR
Singer3	340	√	√	√	√							
lemming	1321	√	√	√	√			√		√		
suv	936				√				√	√		
trellis	561	√	√	√					√		√	
crossing	119		√	√		√		√			√	

4.1.1. Experimental quantitative analysis of the OTB100 dataset

The tracker of this paper's algorithm in OTB100 achieves a superior AUC (0.692) and is compared with 11 trackers namely ToMP, TransT, Ocean, STARK [5], SiamRPN++, ECO, ATOM, SA-Siam [18], GradNet [19], DiMP-50 [20], DaSiamRPN. The results of the one-way evaluation (OPE) curves can be obtained as shown in the Fig. 9.

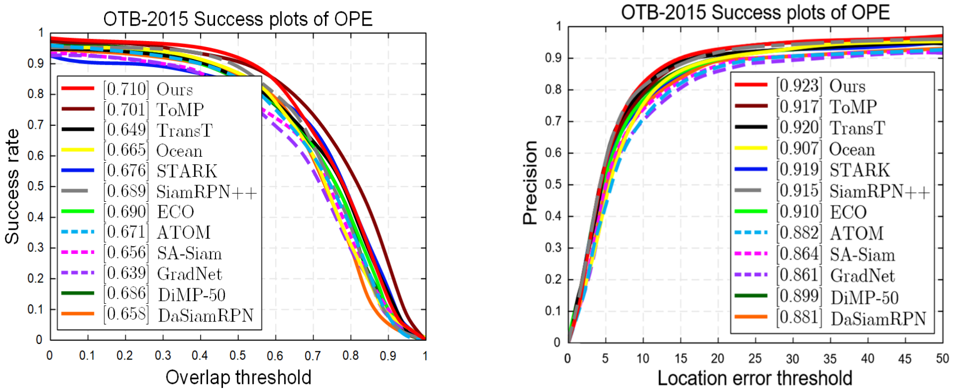


Fig. 9. Results of quantitative analysis of OTB100

To verify the performance of the proposed algorithm in this paper, the same protocol and parameters are used in the quantitative experiments. On the data results in the following Table 3, it can be seen that the algorithm is in the advanced level in terms of precision and accuracy, and that the mixture of fractional-order attention mechanism as well as the improvement of the feature fusion module is a significant contributor to the accuracy enhancement of the tracker. Upon further analysis, the algorithm performs most prominently in the task challenging attributes of OTB100 for sequences with frequent light changes and strong background interference. Due to the attention of FoBAF-T on the target texture features, together with the paper's regression-based discriminative prediction, the tracker is made more robust to target selection, which is specifically reflected in the improvement of 71 % success rate, 92.3 % accuracy rate, and target frame selection is less dependent on the stability of the target state.

Table 3. OTB100 quantification results

Tracker	FoBAF-T	ToMP	TransT	Ocean	STARK	SiamRPN++
Success Rate	0.710	0.701	0.649	0.665	0.676	0.689
Precision	0.923	0.917	0.920	0.907	0.919	0.915
Tracker	ECO	ATOM	SA-Siam	GradNet	DiMP-50	DaSiamRPN
Success Rate	0.690	0.671	0.656	0.639	0.686	0.658
Precision	0.910	0.882	0.864	0.861	0.899	0.881

4.1.2. Experimental quantitative analysis of the LaSOT dataset

Fig. 11 shows the experimental evaluation results of the success rates obtained by SiamRPN, SiamRPN++, ATOM, DiMP, Ocean, TransT, SeqTrack and FoBAF-T among the 11 control algorithms running under 14 challenging attributes of the LaSOT dataset. Fig. 10 visually demonstrates the advancement of the algorithm improvements proposed in this paper in complex scenarios.

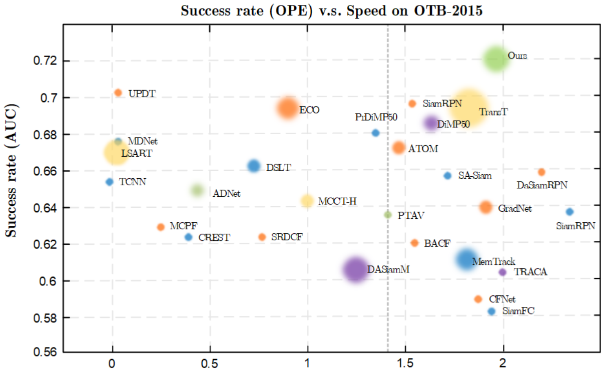


Fig. 10. Comparison of AUC obtained by different algorithms on the OTB100 benchmark

In target tracking, in order to assess the accuracy and reliability of the algorithm, we focus on statistical features such as uncertainty and confidence intervals. Uncertainty assesses the reliability of the tracking results and gives us insight into the potential error and variability of the tracker's predictions. To compute the uncertainty of tracking results, we employ the P and Pnorm metrics to measure the uncertainty in the target tracking task. Specifically, precision is a measure of how accurately the tracker predicts the position of a target object in a given video sequence. It is usually derived by calculating the ratio of correctly predicted frames to the total number of frames. In target tracking, higher accuracy means that the tracker is able to predict the position of the target object more accurately, thus reducing tracking uncertainty. However, accuracy is affected by a variety of factors, such as target occlusion, morphological changes, motion blur, etc., which may lead to a decrease in accuracy, thus increasing tracking uncertainty. Pnorm is a metric that normalizes accuracy by taking into account the effect of the size of the true bounding box. In target tracking, targets of different sizes may lead to different tracking difficulties, so the performance of the tracking algorithm can be evaluated more fairly using Pnorm. A higher value of Pnorm indicates that the tracking algorithm can maintain high accuracy on targets of different sizes, which reduces the tracking uncertainty to a certain extent.

Table 4. LaSOT quantification results

Tracker	FoBAF-T	ToMP	TransT	Ocean	STARK	SiamRPN++
AUC	73.9	68.5	64.9	56.0	67.1	49.6
P_{norm}	82.1	79.2	73.8	65.1	77.0	56.9
P	79.6	73.5	69.0	56.6	-	49.1
Tracker	ECO	ATOM	SeqTrack	TrDiMP	DiMP-50	DaSiamRPN
AUC	32.4	51.5	72.1	63.9	56.9	58.6
P_{norm}	33.8	57.6	81.7	–	65.0	–
P	30.1	50.5	79.0	61.4	56.7	–

In the Table 4, it is evident that FoBAF-T performs slightly better than the latest released algorithm, achieving an AUC score of 73.9 %. The Fig. 11 illustrates the outcomes of the quantitative evaluation conducted under 14 complex scene attributes. This evaluation reveals that the proposed FoBAF-T tracker exhibits a notable advantage in the areas of background discrimination similarity and scale change attribute. This suggests that the model possesses a superior capability in discerning both similarity and scale changes [21-24]. FoBAF-T has a P-value of 73.9 on the ToMP test set, which indicates that it is able to track the target accurately in most cases with low uncertainty. While STARK has a P-value of 56.0 on the SiamRPN++ test set, which indicates that STARK has some errors in the tracking process, and the low P-value increases the uncertainty of target tracking.

4.1.3. Quantitative analysis of other data set experiments

Table 5. Quantification results

Method	Source	TrackingNet			UAV123		VOT		
		AUC	P_{norm}	P	AUC	P	EAO	AUC	SR0.75
FoBAF-T	Ours	86.3	87.9	85.7	70.6	–	30.1	59.7	–
SeqTrack	CVPR2023	85.0	89.5	84.9	68.6	–	–	–	–
ToMP	CVPR2022	81.2	86.2	78.6	66.9	–	29.7	45.3	78.9
TransT	ICCV2021	81.4	86.7	80.3	68.1	87.6	26.6	58.9	35.6
TransT-M	CVPR2022	–	–	–	70.9	–	55.0	74.2	86.9
Ocean	ECCV2020	–	–	–	62.1	–	38.5	58.6	22.0
STARK	ICCV2021	80.3	77.6	85.1	68.6	89.1	68.4	–	–
SiamRPN++	CVPR2019	73.3	80.0	69.4	80.3	61.3	31.5	59.3	35.6
ECO	ICCV2017	55.4	61.8	49.2	52.5	74.1	–	–	–
ATOM	CVPR2019	70.3	77.1	64.8	61.7	82.7	27.1	46.2	73.4
DiMP-50	CVPR2020	74.0	68.7	80.1	65.4	84.9	27.4	45.7	73.4

4.2. Ablation experiment

In this paper, the improvement contributions of different components in FoBAF-T are evaluated through ablation experiment, and the ablation tests are carried out on OTB100 and UAV123. The experimental structure is shown in Table 6.

The components of the ablation experiment are as follows:

HFAF Fractional Hybrid Attention Feature Extraction Network: Here, FoBAF-T modifies ResNet50 and combines it with the channel attention that integrates multi-scale context through replacing the node functions, obtaining a reconstructed feature extraction network. Subsequently, the obtained features are fed into the fusion network.

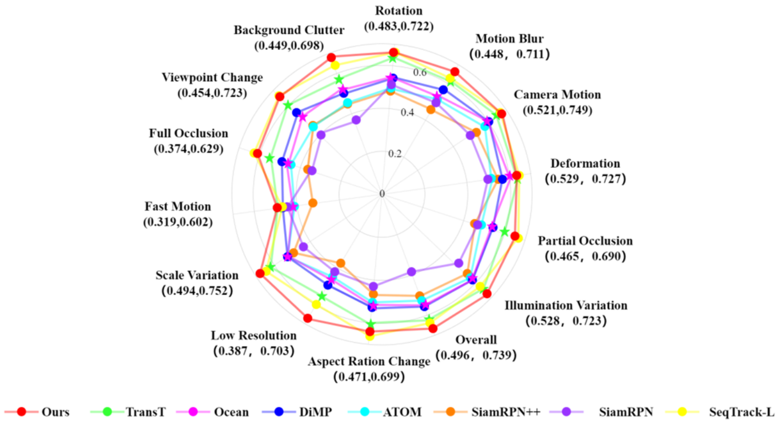


Fig. 11. Comparative results of different challenge attributes on the LaSOT dataset

FoBAF Feature Fusion Network: Compared to the baseline algorithm, the fundamental difference between the feature fusion network in this paper lies in the node function design and the strategy of Siamese branch fusion, which is decomposed into a two-stage structure with a focus on the contribution of the two methods to the similarity measure in the algorithm. The experimental data obtains in Table 5 indicates that if the features are directly weighted and summed without any other processing, the desired result of the algorithm in this paper cannot be achieved [25-27].

Table 6. Ablation experiments based on LaSOT and UAV123

Method		FoBAFT-nCA	FoBAFT-nf	FoBAFT-nfo	FoBAFT-np
LaSOT	AUCP	68.8	64.6	70.9	73.1
		78.4	77.8	78.5	79.3
UAV123	AUCP	61.8	60.6	63.7	69.2
		83.9	84.1	86.0	89.1

Table 7. Ablation experiments based on LaSOT and TrackingNet

Method	FoMCA	FoBAF	Correlation	LaSOT		TrackingNet	
				AUC	*P*	AUC	*P*
A	–	–	√	0.487	0.496	0.571	0.532
A	√	–	√	0.509	0.503	0.583	0.541
A	–	√	–	0.515	0.509	0.599	0.549
A	–	√	√	0.511	0.507	0.593	0.542
B	–	–	√	0.512	0.499	0.686	0.648
B	√	–	√	0.520	0.510	0.713	0.652
B	√	√	–	0.523	0.519	0.749	0.687

4.3. Comparative experiment

To visually observe the robustness of the algorithms in this paper under a variety of difficult scenario factors, the experiments in this subsection selected the visualization test results of the algorithms in this paper and the comparison algorithms on randomly selected sequences in three datasets, namely, OTB100, LaSOT, UAV123, and ImageNet. This subsection presents four video sequences. The sequence of tracking frames is represented from left to right, respectively. Among them, the green bounding box represents the real tagged value results; the red bounding box represents the tracking prediction results of the algorithm in this paper; the light blue bounding box represents the DiMP algorithm tracking prediction results; the dark blue bounding box represents the SeqTrack tracking prediction results; the pink bounding box represents the SiamRPN tracking prediction results; and the yellow bounding box represents the baseline algorithm TransT tracking prediction results. The above qualitative analysis is shown below in Fig. 12.

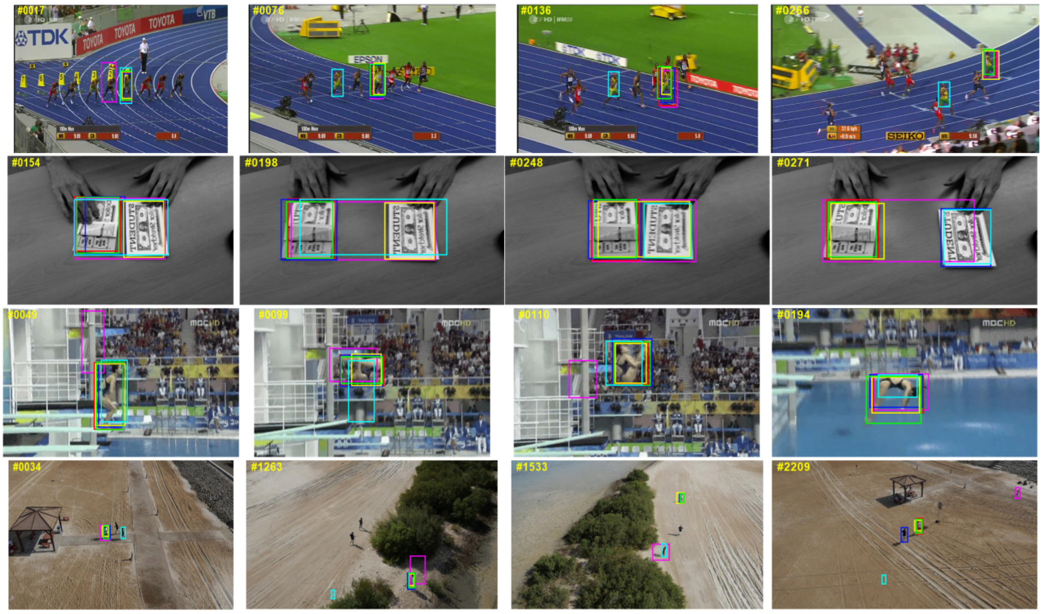


Fig. 12. Results of qualitative analysis under randomized sequences of four test sets

It can be observed from the following experimental results that in the bolt sequence involving challenges of fast movement and deformation, when the target is occluded by similar backgrounds in close proximity around it, the tracking boxes of both the SiamRPN and DiMP algorithms exhibit drift and tracking identity errors. However, compared with the compared algorithms Ours-A and FoFTr-T, the tracking results of the algorithm presented in this paper are more stable. In the Diving challenge involving fast motion and motion blur, the tracking bounding box of SiamRPN drifts due to the cluttered background. When there are drastic shape changes and complex backgrounds around frame 99, DiMP suffers from severe drift and inaccurate tracking box size. The algorithm presented in this paper is stable and capable of adapting to the rotation and deformation of the target. In the Group2 sequence that involves fast movement of small targets and scale variations, the algorithm in this paper yields tracking results that are more in line with expectations compared to the baseline algorithm TransT and the advanced tracker SeqTrack.

5. Conclusions

This paper primarily designs and implements target tracking based on fractional-order bidirectional feature fusion to solve the tracking drift and tracking frame jumping in complex scenes. Firstly, the starting point and overall structure of the model design are introduced. Secondly, in view of the low information utilization rate of the feature extraction network in the target tracking framework at the local scale of the target, an attention-CNN hybrid attention extraction module based on the fractional order node function is proposed to enhance the robustness of target tracking. Then the feature fusion network is improved to achieve enhancement and discriminative enhancement of the hybrid features. Eventually, the overall performance is evaluated quantitatively and qualitatively on multiple evaluation datasets with a variety of state-of-the-art trackers. In target tracking, in order to evaluate the accuracy and reliability of the algorithm, we focus on statistical features such as uncertainty, which assesses the reliability of the tracking results, and confidence intervals, which provide a range of confidence for the tracking results. These features are crucial for understanding the performance and limitations of the algorithm. From the results, it can be seen that the algorithm proposed in this paper has considerable tracking advantages under drastic changes in scale patterns, dynamic blurring, and similar interference properties.

Acknowledgements

The authors have not disclosed any funding.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Author contributions

Yao Fu: conceptualization, methodology, validation, writing-original draft preparation, writing-review and editing. Yilu Wang: conceptualization, methodology, validation.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, pp. 1137–1149, Jun. 2017, https://doi.org/10.1109/tpami.2016.2577031

[2] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High performance visual tracking with Siamese region proposal network,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, https://doi.org/10.1109/cvpr.2018.00935

[3] Z. Chai, Y. Ling, Z. Luo, D. Lin, M. Jiang, and S. Li, “Dual-stream transformer with distribution alignment for visible-infrared person re-identification,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 33, No. 11, pp. 6764–6776, Nov. 2023, https://doi.org/10.1109/tcsvt.2023.3268080

[4] Q. Wei, B. Zeng, J. Liu, L. He, and G. Zeng, “LiteTrack: layer pruning with asynchronous feature extraction for lightweight and efficient visual tracking,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4968–4975, May 2024, https://doi.org/10.1109/icra57147.2024.10610022

[5] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, “Learning spatio-temporal transformer for visual tracking,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10428–10437, Oct. 2021, https://doi.org/10.1109/iccv48922.2021.01028

[6] L. Lin, H. Fan, Z. Zhang, Y. Xu, and H. Ling, “SwinTrack: a simple and strong baseline for transformer tracking,” *arXiv*, Jan. 2021, https://doi.org/10.48550/arxiv.2112.00995

[7] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, “Transformer tracking,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, https://doi.org/10.1109/cvpr46437.2021.00803

[8] Y. Cao, W. Wang, Y. Zhao, and Q. Sun, “Research on visual inspection method of tree whitening quality based on multi-level feature fusion,” *Recent Patents on Engineering*, Vol. 18, No. 3, p. e080523216685, Apr. 2024, https://doi.org/10.2174/1872212118666230508163955

[9] C. Liqun and Shiqi, “Research on anti-obscure target tracking method based on feature adaptive fusion,” in *IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, pp. 1588–1592, Feb. 2023, https://doi.org/10.1109/eebda56825.2023.10090759

[10] K. Xiao and Z. Hao, “Research on target tracking algorithm based on multi-layer convolution feature fusion,” in *International Conference on Industrial Automation, Robotics and Control Engineering (IARCE)*, pp. 1–8, Jun. 2022, https://doi.org/10.1109/iarce57187.2022.00011

[11] Y. Zhang and M. Dong, "Research on visual SLAM algorithm based on improved point-line feature fusion," in *International Conference on Pattern Recognition, Machine Vision and Intelligent Algorithms (PRMVIA)*, pp. 245–251, Mar. 2023, https://doi.org/10.1109/prmvia58252.2023.00046

[12] Z. Shi, M. Chen, and Z. Wu, "Hyperspectral image classification based on dual-scale dense network with efficient channel attentional feature fusion," *Electronics*, Vol. 12, No. 13, p. 2991, Jul. 2023, https://doi.org/10.3390/electronics12132991

[13] S. Wan, T. Li, B. Fang, K. Yan, J. Hong, and X. Li, "Bearing fault diagnosis based on multisensor information coupling and attentional feature fusion," *IEEE Transactions on Instrumentation and Measurement*, Vol. 72, pp. 1–12, Jan. 2023, https://doi.org/10.1109/tim.2023.3269115

[14] J. Qu, C. Tang, Y. Zhang, K. Zhou, and A. Razi, "Long-time target tracking algorithm based on re-detection multi-feature fusion," *IET Cyber-Systems and Robotics*, Vol. 4, No. 1, pp. 38–50, Mar. 2022, https://doi.org/10.1049/csy2.12042

[15] M. Zhao, Q. Yue, D. Sun, and Y. Zhong, "Improved SwinTrack single target tracking algorithm based on spatio-temporal feature fusion," *IET Image Processing*, Vol. 17, No. 8, pp. 2410–2421, Apr. 2023, https://doi.org/10.1049/ipr2.12803

[16] S. Chan, J. Tao, X. Zhou, B. Wu, H. Wang, and S. Chen, "Target tracking based on standard hedging and feature fusion for robot," *Industrial Robot: the international journal of robotics research and application*, Vol. 48, No. 5, pp. 659–672, Sep. 2021, https://doi.org/10.1108/ir-09-2020-0212

[17] Y. Yang and X. Gu, "Joint correlation and attention based feature fusion network for accurate visual tracking," *IEEE Transactions on Image Processing*, Vol. 32, pp. 1705–1715, Jan. 2023, https://doi.org/10.1109/tip.2023.3251027

[18] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold Siamese network for real-time object tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4834–4843, Jun. 2018, https://doi.org/10.1109/cvpr.2018.00508

[19] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: gradient-guided network for visual object tracking," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6161–6170, Oct. 2019, https://doi.org/10.1109/iccv.2019.00626

[20] X. He and C. Y.-C. Chen, "Learning object-uncertainty policy for visual tracking," *Information Sciences*, Vol. 582, pp. 60–72, Jan. 2022, https://doi.org/10.1016/j.ins.2021.09.002

[21] I. Paglianti and A. Cristofaro, "Fault-tolerant formation control of wheeled mobile robots using energy-balancing methods," in *European Control Conference (ECC)*, Vol. 27, pp. 472–477, Jul. 2022, https://doi.org/10.23919/ecc55457.2022.9838156

[22] G. Liu, J. H. Park, H. Xu, and C. Hua, "Reduced-order observer-based output-feedback tracking control for nonlinear time-delay systems with global prescribed performance," *IEEE Transactions on Cybernetics*, Vol. 53, No. 9, pp. 5560–5571, Sep. 2023, https://doi.org/10.1109/tcyb.2022.3158932

[23] J.-X. Zhang and G.-H. Yang, "Low-complexity tracking control of strict-feedback systems with unknown control directions," *IEEE Transactions on Automatic Control*, Vol. 64, No. 12, pp. 5175–5182, Dec. 2019, https://doi.org/10.1109/tac.2019.2910738

[24] J.-X. Zhang and G.-H. Yang, "Fuzzy adaptive output feedback control of uncertain nonlinear systems with prescribed performance," *IEEE Transactions on Cybernetics*, Vol. 48, No. 5, pp. 1342–1354, May 2018, https://doi.org/10.1109/tcyb.2017.2692767

[25] J.-X. Zhang, Q.-G. Wang, and W. Ding, "Global output-feedback prescribed performance control of nonlinear systems with unknown virtual control coefficients," *IEEE Transactions on Automatic Control*, Vol. 67, No. 12, pp. 6904–6911, Dec. 2022, https://doi.org/10.1109/tac.2021.3137103

[26] Y. Wang, Q. Yang, L. Liu, and X. Zhang, "A cross-domain few-shot visual object tracker based on bidirectional adversary generation," *IEEE Sensors Journal*, Vol. 24, No. 12, pp. 19506–19516, Jun. 2024, https://doi.org/10.1109/jsen.2024.3394525

[27] Q. Yang, Y. Wang, L. Liu, and X. Zhang, "Adaptive fractional-order multi-scale optimization TV-L1 optical flow algorithm," *Fractal and Fractional*, Vol. 8, No. 4, p. 179, Mar. 2024, https://doi.org/10.3390/fractalfract8040179

Yao Fu received the M.S. degree in Communication and Information System in Shenyang Ligong University in 2015. She is a senior engineer in the School of Information Science and Engineering. Her research interests include wireless sensor networks and routing protocol design, digital image processing and single chip.

Yilu Wang received the B.S. degree in vehicle engineering from the Shenyang Ligong University, Shenyang, China, in 2021. From 2021 onwards, she has been pursuing the master's degree in mechanical and electronic engineering. Her research interests include deep learning, fractional differential, visual object tracking, and image super resolution. She is currently studying at the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China. Her current research interests include fractional-order control and intelligent control.