Repformer: A Robust Shared-Encoder Dual-Pipeline Transformer for Visual Tracking

Fengwei Gu, Jun Lu*, Chengtao Cai, Qidan Zhu, and Zhaojie Ju*

Abstract—Siamese-based trackers have achieved outstanding tracking performance. However, these trackers in complex scenarios struggle to adequately integrate the valuable target feature information, which results in poor tracking performance. In this paper, a novel shared-encoder dual-pipeline Transformer architecture is proposed to achieve robust visual tracking. The proposed method integrates several main components based on a hybrid attention mechanism, namely the shared encoder, the feature enhancement pipelines with functional complementarity, and the pipeline feature fusion head. The shared encoder is adopted to process template features and provide useful target feature information for the feature enhancement pipeline. The feature enhancement pipeline is responsible for enhancing feature information, establishing feature dependencies between the template and the search region, and employing global information adequately. To further correlate the global information, the pipeline feature fusion head integrates the feature information from the feature enhancement pipelines. Eventually, we propose a robust Siamese-based Repformer tracker, which incorporates a concise tracking prediction network to obtain efficient tracking representations. Experiments show that our tracking method surpasses numerous state-of-the-art trackers on multiple tracking benchmarks, with a running speed of 57.3 fps.

Keywords—Visual tracking, Transformer architecture, hybrid attention mechanism, pipeline feature fusion head.

1. Introduction

Visual tracking is an extremely meaningful research content in the field of computer vision, which encompass a very wide range of application scenarios, such as unmanned driving [1,2], human-computer interaction [2,3,6], and intelligent monitoring [3,4]. However, practical applications often involve numerous complex scenarios Therefore, the tracking process is often accompanied by interference factors [5] such as occlusion, fast motion, and scale variation [3, 4,7], which seriously limit the tracking accuracy. Despite notable advancements made by some trackers [4-8], visual tracking in complex scenarios remains a challenging task.

For continuous video sequences, the purpose of visual tracking [8] is to locate the target contained in the subsequent frames, given the initial bounding box of the target in the first frame [9, 10]. Currently, popular visual tracking methods [1, 5-14] can be roughly divided into two categories: the correlation filtering-based tracking methods [1, 5, 7, 13, 14] and the deep learning-based tracking methods [3, 6, 8-12]. Generally speaking, the correlation filtering-based tracking methods habitually employ MOSSE [13] to obtain the correlation filtering results to complete the tracking process through the minimum optimization approach. Moreover, to improve the accuracy of the correlation filtering-based tracking methods [5,7,14], additional measures [14,15,18-21] need to be taken to obtain more target feature information. Typical measures [14] generally fall into two types: one is to apply kernel techniques in the correlation filtering-based tracking methods, such as KCF [14] and CSK [15] trackers; the other is to use deep convolutional neural networks as feature extractors to obtain deep features [24] through offline learning. To solve the problem of low tracking efficiency [18-21], some trackers have opted to abandon the model refresh mechanism [22]. However, this approach seriously compromises the accuracy of the tracker. Therefore, how to balance accuracy and efficiency is particularly critical for trackers [36]. Recently, deep learning-based tracking methods [8-12] have garnered increasing attention from researchers due to their notable advantages in tracking accuracy [3]. However, as the complexity of the model itself increases, it adversely impacts the tracking speed [12]. After the emergence of the Siamese-based tracker [25], it has alleviated the contradiction between tracking accuracy and speed [11] to a certain extent, while achieving a favorable balance between the two [10-12]. In the composition of the Siamese-based tracker [25], the correlation operation is responsible for processing

This work is supported by the Natural Science Foundation of Heilongjiang Province of China under Grant No. F201123, the National Natural Science Foundation of China under Grant 52171332 and 52075530, the Green Intelligent Inland Ship Innovation Programme under Grant MC-202002-C01, and the Development Project of Ship Situational Intelligent Awareness System under Grant MC-201920-X01. (*Corresponding author: Jun Lu; Zhaojie Ju)

Fengwei Gu, Jun Lu, Chengtao Cai, and Qidan Zhu are with the College of Intelligent Systems Science and Engineering and the Key Laboratory of Intelligent Technology and Application of Marine Equipment, Ministry of Education, Harbin Engineering University, Harbin 150001, China (e-mail: gufengwei@hrbeu.edu.cn; lujun_larry@hrbeu.edu.cn; caichengtao@hrbeu.edu.cn; zhuqidan@hrbeu.edu.cn).

Zhaojie Ju is with the School of Computing, University of Portsmouth, Portsmouth, PO1 3HE, UK (e-mail: Zhaojie.Ju@port.ac.uk).

target features from the template and search region [27] to obtain response results. Video sequences contain particularly rich feature dependencies and feature information [36]. If the relevant feature information can be efficiently utilized [66] and the feature dependencies can be fully mined [67], the performance of the tracker can be significantly improved. However, the correlation operation only focuses on the local information [66] and ignores the global information, which is difficult to establish outstanding feature dependencies. It is worth noting that most trackers employing correlation operations [8, 10, 59, 67] struggle to efficiently integrate the feature information from the template and search region [10, 59], leading to the loss of valuable feature information. Unfortunately, the correlation filtering-based trackers [5,7,13,14] encounter a similar situation. Furthermore, especially in complex scenes, most trackers fail to efficiently and fully utilize the useful feature information [7,10,13,14,59,66], which results in a sharp decline in the tracking effect [34]. Therefore, many popular trackers incorporate additional structures [5] and refresh modules [5, 8] to improve the utilization level of feature information. With the emergence of the Transformer, researchers have been presented with a broader avenue for exploration.

To solve the above-mentioned key problems, a robust tracking framework is proposed to complete the tracking representation in complex scenarios. The main contributions of our method can be summarized as follows:

1) We propose a tracker based on the ideas of Transformer and Siamese, named Repformer, to achieve robust visual tracking in complex scenes. Our method only employs a concise tracking prediction network to achieve the prediction and localization of the tracking bounding box, without relying on prior information or excessive hyperparameter adjustments. The proposed Repformer can acquire outstanding tracking performance in complex environments.

2) We develop a novel shared-encoder dual-pipeline Transformer architecture based on a hybrid attention mechanism, which consists of the shared encoder, the feature enhancement pipelines (P1, P2), and the pipeline feature fusion head. The designed shared encoder simultaneously connects the feature enhancement pipelines P1 and P2, which have complementary functions. It delivers the processed template features to the P1 and P2 pipelines, enabling interaction with the search region features and achieving feature enhancement. Finally, the proposed pipeline feature fusion head integrates the output features from the P1 and P2 pipelines to facilitate the correlation of global information. In this process, the feature information between the template and the search region can be sufficiently integrated and enhanced. Meanwhile, rich feature dependencies and nonlinear relationships are also obtained.

3) To illustrate the superiority of our method, the proposed tracker achieves outstanding performance on 7 challenging datasets while running at 57.3fps. Meanwhile, our method outperforms the reported typical tracking methods on 14 challenging attributes.

The rest of the paper is organized as follows: representative work in the field of visual tracking is introduced in Section 2, and the proposed Repformer method is presented in Section 3. Experiments and discussions are conducted in Sections 4 and 5, respectively. Ultimately, the conclusion in this paper is revealed in Section 6.

2. Related Work

2.1. Visual tracking methods

Currently, the visual tracking methods involved can be broadly classified into two categories: one is the correlation filtering-based tracking methods [1,5,7,13,14,17,18], and the other is the deep learning-based tracking methods [3,6,8-12,20,59,66,67]. In the correlation filtering-based tracking methods [17], the primary concept is to evaluate the correlation [13,14] between the template image and the input video image. MOSSE [13] first applies correlation filtering in its tracking architecture [17] and implements a model refresh strategy. Although MOSSE [13] exhibits average tracking accuracy, it shows excellent robustness and running speed against disturbances such as scale variation, attitude, and illumination changes [18]. These characteristics hold valuable reference significance for subsequent tracking methods. To mitigate the interference caused by boundary effects, ASTCA [73] utilizes spatial-temporal context-aware weights to distinguish the target from the background in tracking tasks, while incorporating spatial context information to reduce background interference. ALT [74], on the other hand, employs an active learning approach by selecting and annotating unlabeled samples for training the tracking model. It also uses a nearest neighbor discrimination method to filter out inappropriate samples. Additionally, the Tversky loss is adopted to enhance tracking accuracy.

AMFT [75] integrates different features and employs a model update strategy to accommodate target changes. To address the tracking challenges in complex environments and improve the tracking accuracy [14], the correlation filtering-based tracking methods [15] incorporate a kernel learning technique, such as CSK [15] and KCF [14]. CSK [15] adopts a cyclic structure to generate the response map of the target feature to accomplish the tracking. Building upon CSK [15], KCF [14] combines a Gaussian kernel function to extract the target from video sequences and employs the HOG descriptor for feature representation. Due to KCF [14], SAMF [17] utilizes a multi-feature fusion method and a multi-scale search strategy to achieve adaptive tracking, which improves the tracking accuracy to a certain extent. Owing to the excellent performance of the kernel learning technique [14,15,17], it has been generalized and applied to numerous tracking methods, such as CFNet [18], Staple [19], CCOT [20], ECO [21], UPDT [22], SRDCF [23], and DeepSRDCF [24]. It can be noted that some aforementioned tracking methods, such as CFNet [18], CCOT [20], and ECO [21], leverage convolutional neural networks to extract more informative target features. As a result, research interest has gradually shifted towards deep learning-based tracking methods, aiming for higher performance improvements.

Among the deep learning-based tracking methods [3,6,8-12,20,59,66,67], the Siamese-based tracker [8,10,59,66] stands out with its exceptional performance, which obtains a relative balance [59,66] between tracking accuracy and speed. SiamFC [25] combines a Siamese architecture and a fully convolutional network to calculate the correlation [10] between the template and search region [8], enabling lightweight visual tracking without the need for a refresh module. Although SiamFC has a simple network structure [25], which results in outstanding tracking speed, its tracking accuracy [20] is relatively lower compared to subsequent state-of-the-art trackers [59,66,67]. Nevertheless, SiamFC [25] has garnered significant attention due to its pioneering research results. DSiam [26] exploits an online learning approach to deal with the appearance changes of the tracking target, while resisting background information and achieving dynamic tracking. To obtain excellent tracking performance, SiamRPN [27] proposed by Li et al. combines the Siamese architecture [25] with the RPN network. It employs classification and regression branches for accurate tracking. Reference [27] shows that SiamRPN can ensure both tracking accuracy and running speed, which indicates that SiamRPN has expansive application prospects. The proposed DaSiamRPN [28] further improves the recognition ability of the tracking network [10], thanks to the application of negative samples. Moreover, it utilizes an interference-aware module [28] and a local-to-global search method to improve tracking efficiency. SiamCorners [76] interprets the estimation of bounding boxes as predictions of corners, which avoids the need for anchor box design and makes the tracking framework more flexible. During tracking, Self-SDCT [57] utilizes a pre-trained feature extraction network to extract features and locates the tracking target using the Siamese correlation tracking architecture. Additionally, numerous excellent tracking methods have also attracted wide attention, including SiamRPN++ [29], SiamR-CNN [30], SiamCAR [31], etc. It is worth mentioning that some improvements can be introduced into the Siamese-based tracking methods, containing representation learning [18], additional modules [5, 8], and advanced backbone networks [21], among others.

Recently, some outstanding tracking methods have continued to emerge, such as HCAT [8], SiamCAN [59], SiamPCF [10], TRAT [60], CSVMF [64], TransT [66], STARK [67], BAASR [5], STMTrack [9], UPDT [22], KYS [32], ATOM [33], and Ocean [34]. These trackers have achieved state-of-the-art performance, along with the potential for further development and application.

Although the aforementioned tracking methods have achieved prominent performance [20,59,66,67], there is still significant room for improvement in utilizing the feature information of the tracking target in video sequences. Especially in complex environments [66], if the useful information from target features can be fully utilized [59], the tracking performance will be greatly improved. Therefore, a dual-pipeline Transformer framework with a shared encoder based on a hybrid attention mechanism is developed to efficiently utilize target feature information and reduce information loss, which allows the target features between the template and search region to be fused and enhanced sufficiently.

2.2. Transformer

The Transformer, originally proposed in [16], is a practical attention-based network framework. The attention module serves as a crucial component within the Transformer [16], which constitutes the core architecture of the encoder and decoder [59] in the main structure. Besides, the Transformer [16] can process sequence information well [8] and obtain the dependencies of each element in the sequences [66], which

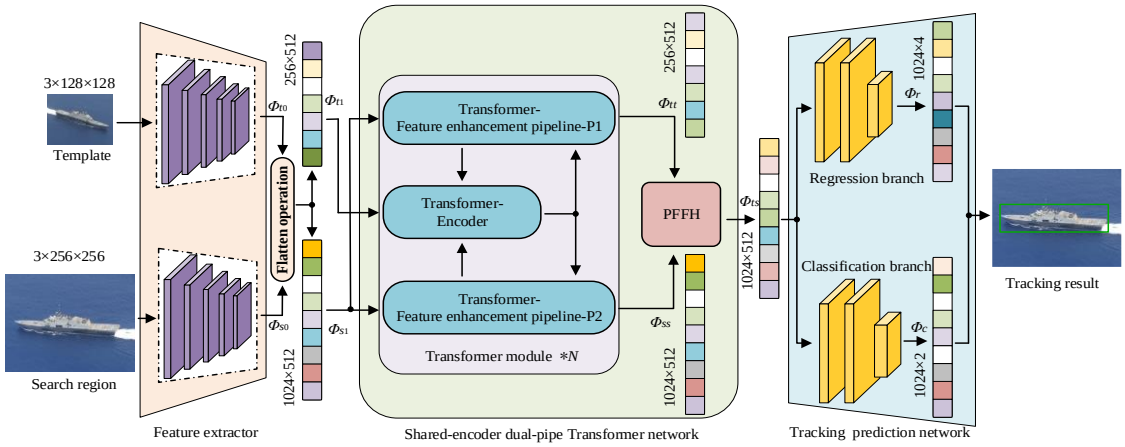

Fig. 1. The overall tracking framework of the proposed method. Our framework is particularly concise, which consists of three basic elements. In addition, PFFH represents the pipeline feature fusion head. *N* indicates the number of modules. The remaining symbols and numbers stand for feature vectors and sizes, respectively.

motivates itself to capture global information [8,16,59,66] from the sequences. In the field of natural language processing [35], the Transformer has demonstrated remarkable superiority due to its exceptional memory mechanism and powerful computational capabilities [59], thereby attracting significant attention. Especially in the direction of machine translation, the Transformer has become a considerably popular method.

Recently, the Transformer and attention mechanism [16] have been introduced into many typical visual tasks [8,59,66], such as semantic segmentation, image classification, and target tracking [8,36,37]. It can be noted that the Transformer and its variant structures [8,38,59] are also employed in the discontinuous work, which still obtains outstanding performance. Therefore, it is meaningful to further explore the applications of the Transformer in related fields. Meanwhile, at this stage, it is still particularly necessary to study how to fully exploit the potential of the Transformer for visual tracking.

In this paper, we do not directly adopt the original framework of Transformer [16]. Instead, we borrow the main idea of Transformer [16] and combine it with the Siamese network system [25]. Furthermore, we redesign a robust Transformer variant based on a hybrid attention mechanism, incorporating a shared encoder and two functionally complementary feature enhancement pipelines to enable feature interaction. Meanwhile, the pipeline feature fusion head, based on a hybrid attention mechanism, is specially designed to fuse and enhance features, which establishes the final association of target features between the template and search region. Therefore, a novel Transformer tracker is proposed to efficiently integrate feature information for robust target tracking in complex scenes.

3. Proposed Repformer Method

In this section, the proposed Repformer will be introduced in detail. Firstly, the overall tracking framework of the proposed method is described, which clarifies the roles and details of each component. Then, the proposed shared-encoder dual-pipeline Transformer network is specifically analyzed to present the inference flow of the network architecture for target feature information. Finally, the training loss of our method is presented.

3.1. Overall tracking framework

A simple Transformer-based Siamese tracker [16, 25] is proposed, named Repformer, to achieve a robust tracking representation. The overall tracking framework of our method is shown in Fig. 1, which mainly consists of three components: a feature extractor, a shared-encoder dual-pipeline Transformer network, and a tracking prediction network.

In our tracking framework, image pairs are adopted as the input of the feature extractor. These pairs consist of the template image $T \in \mathbb{R}^{3\times H_t\times W_t}$ and the search region image $S \in \mathbb{R}^{3\times H_s\times W_s}$, where 3 represents the corresponding channel number. In addition, H_i and W_i ($i = t, s$) denote the height and width, respectively.

It can be mentioned that we have the flexibility to choose any convolutional neural network as the feature extractor for extracting target features from the template and search region. Here, ResNet50 [39] is employed as the feature extractor for the proposed method. However, instead of applying the original ResNet50 [39] directly, we make minor changes to it. Specifically, the last convolution stage and fully connected layer in the ResNet50 network [39] are removed and the down-sampling stride is set to 1 in the fourth convolution stage to improve feature resolution. After passing through the feature extractor, the template image T and the search region image S are mapped to obtain the corresponding feature maps $\Phi_t \in \mathbb{R}^{C\times H_{t1}\times W_{t1}}$ and $\Phi_s \in \mathbb{R}^{C\times H_{s1}\times W_{s1}}$. Here, H_j and W_j (j=t1, s1) denote the corresponding height and width, respectively, and C represents the channel number in the feature map. Before entering the next-level processing unit, the dimensions of the feature maps Φ_t and Φ_s need to be reduced to D, to obtain the corresponding feature maps $\Phi_{t0} \in \mathbb{R}^{D\times H_{t1}\times W_{t1}}$ and $\Phi_{s0} \in \mathbb{R}^{D\times H_{s1}\times W_{s1}}$. Subsequently, to meet the input requirements of the proposed Transformer, a flatten operation on Φ_{t0} and Φ_{s0} is performed to generate the corresponding template feature vector $\Phi_{t1} \in \mathbb{R}^{D\times H_{t1}W_{t1}}$ and search region feature vector $\Phi_{s1} \in \mathbb{R}^{D\times H_{s1}W_{s1}}$.

The proposed shared-encoder dual-pipeline Transformer architecture is used to efficiently integrate the target features from the template and search region, to achieve the fusion and enhancement of the target features. Specifically, the template feature vector Φ_{t1} and the search region feature vector Φ_{s1} from the feature extractor are sufficiently processed to achieve the purpose of global feature association. The Transformer we designed has three input ports, which correspond to the shared encoder port, the feature enhancement pipeline P1 port, and the P2 port, respectively. Among them, the shared encoder port is responsible for receiving the template feature vector Φ_{t1}, while both the feature enhancement pipeline P1 port and P2 port receive the search region feature vector Φ_{s1}. When the feature vectors pass through the proposed shared encoder and feature enhancement pipelines (P1, P2), the corresponding two integrated output vectors $\Phi_{tt} \in \mathbb{R}^{D\times H_{t1}W_{t1}}$ and $\Phi_{ss} \in \mathbb{R}^{D\times H_{s1}W_{s1}}$ can be obtained, which contain rich target feature information and provide favorable conditions for tracking and locating the target. Then, the feature vectors Φ_{tt} and Φ_{ss} flow into the pipeline feature fusion head to acquire the fused feature vector $\Phi_{ts} \in \mathbb{R}^{D\times H_{s1}W_{s1}}$, which is the final output of our Transformer. Moreover, the specific content of our proposed Transformer is detailed in Section 3.2. It should be noted that the proposed Transformer is the core element in our tracking method and occupies a rather important position.

The tracking prediction network in the proposed method is responsible for predicting and localizing the tracking bounding box in the video sequences. To simplify the tracking process, our tracking prediction network exclusively adopts a multilayer perceptron to further process the feature vector $\Phi_{ts} \in \mathbb{R}^{D\times H_{s1}W_{s1}}$ from the proposed Transformer. Specifically, the feature vector Φ_{ts} flows through two parallel branches in the tracking prediction network, namely the classification branch and the regression branch. Then, the corresponding feature vectors $\Phi_r \in \mathbb{R}^{reg\times H_{s1}W_{s1}}$ and $\Phi_c \in \mathbb{R}^{cla\times H_{s1}W_{s1}}$ from the classification branch and regression branch can be obtained, where *reg* and *cla* represent the output dimension of the corresponding branch, respectively. At this point, the tracking bounding box of the target in the video sequences can be determined. It should be noted that the two parallel branches are composed of 3-layer perceptrons, and the dimensions of the input layer and the intermediate layer are both 512. Multilayer perceptrons enable our tracking prediction network to improve tracking efficiency without relying heavily on prior information or requiring extensive tuning of hyperparameters.

3.2. Proposed Transformer Network

The dual-pipeline Transformer network we designed with a shared encoder can efficiently integrate feature information from the template and search region, while effectively establishing global dependencies between features. The proposed Transformer network consists of several core components: the shared encoder, the feature enhancement pipelines (P1, P2), and the pipeline feature fusion head. In addition, the above-mentioned components all employ a hybrid attention mechanism.

3.2.1. Attention mechanism

The attention mechanism occupies a particularly important position in the proposed Transformer [16] network, which is the key to fully utilize the target feature information. As shown in Fig. 1, the attention mechanism [16] is widely present in the constituent elements of our Transformer. When the input q (queries), k (keys), and v (values) are presented, the attention function is obtained, which can be defined as the Eq. (1).

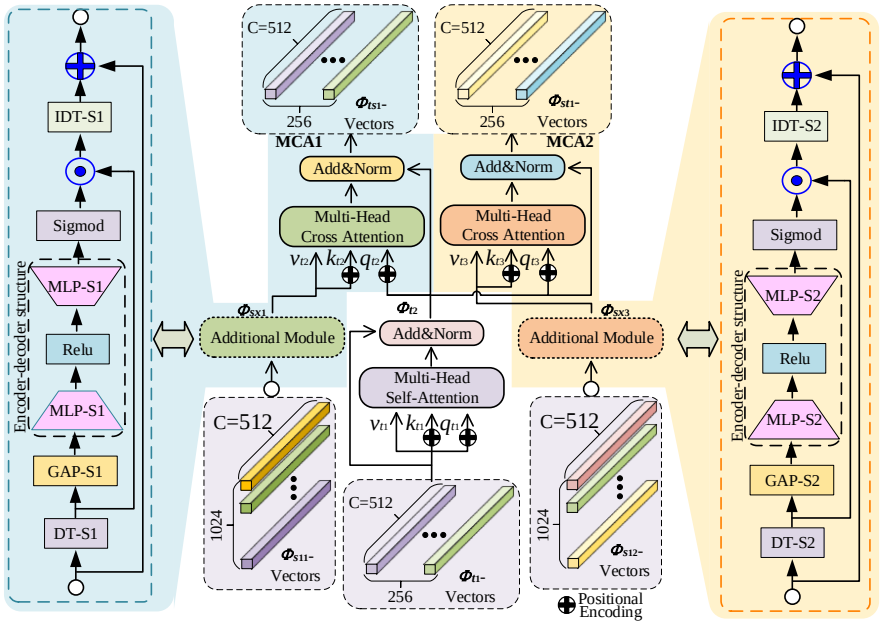

Fig. 2 The network architecture of the shared encoder. To provide location information, positional encodings are added to the network architecture. Moreover, the feature processing capability of the network can be enhanced by additional modules.

To make the model obtain more abundant feature information, the multi-head attention mechanism based on Eq. (1) emerges as a pivotal solution, as shown in Eqs. (2) and (3). The hybrid attention mechanism involved in this paper is inspired by this point.

$$Attention(q,k,v) = softmax\left(qk^T/\sqrt{D_k}\right)v \tag{1}$$

$$H_i = Attention\left(qW_i^q, kW_i^k, vW_i^v\right) \tag{2}$$

$$MultiHead(q,k,v) = Concat(H_1, H_2, \dots, H_h)W^O \tag{3}$$

where D_k indicates the dimension of key value k, and $i \in [1,h]$. Besides, $W_i^q \in \mathbb{R}^{D_m \times D_k}$, $W_i^k \in \mathbb{R}^{D_m \times D_k}$, $W_i^v \in \mathbb{R}^{D_m \times D_v}$, and $W^O \in \mathbb{R}^{hD_v \times D_m}$ represent parameter matrices. It should be noted that D_m and D_v have the identical meaning as D_k.

3.2.2. Shared Encoder

As shown in Fig. 2, the network architecture of the shared encoder is presented. Our shared encoder takes the feature vector Φ_{t1} corresponding to the template image as the input, together with two parallel shunt outputs Φ_{ts1} and Φ_{st1}. Furthermore, additional modules in the network can be any type of modules with the feature processing capabilities, such as feature enhancement, which is not restricted. In view of this, we adopt a channel feature selection module instead of an additional module for selecting features of interest. Specifically, we obtain channel descriptions through data transformation (DT) and global average pooling (GAP). Subsequently, an encoder-decoder architecture is used to obtain inter-channel dependencies. Finally, through dot product and inverse data transformation (IDT), we can obtain features of interest for further processing. To simplify the network architecture of the shared encoder as much as possible, we directly utilize the output Φ_{sx1} and Φ_{sx3} of the proposed additional modules, without using other additional modules. It can be seen that the proposed shared encoder consists of a multi-head self-attention submodule and two multi-head cross-attention submodules. All submodules involved are configured with the residual structure followed by layer normalization, which facilitates the robust training of the overall model. Therefore, the output of each submodule can be generalized as $output = LayerNorm\big(input + submodel(input)\big)$, where $submodel(\cdot)$ represents the submodule function and $LayerNorm(\cdot)$ denotes the layer normalization function. In addition, it can be seen from Fig. 1 that when the number of modules is N=1, the reasoning process of the shared encoder mechanism can be summarized as follows:

In the shared encoder, the inputs q_{t1}, k_{t1}, and v_{t1} of the multi-head self-attention submodule are all the

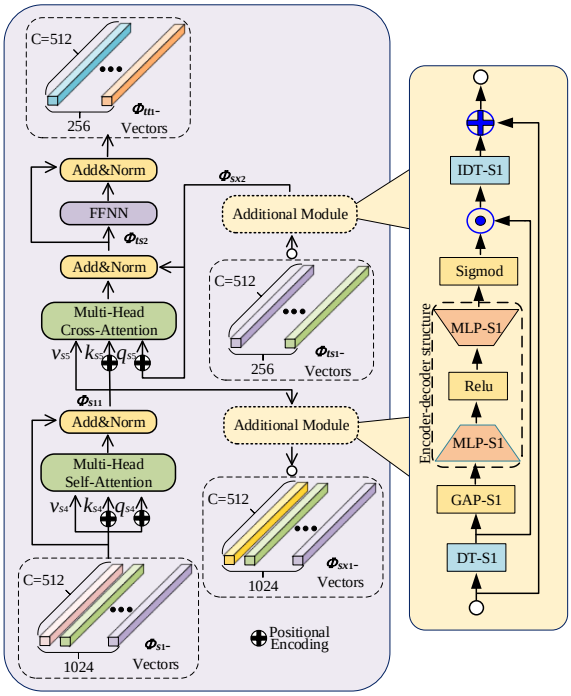

Fig. 3 The network structure of the feature enhancement pipeline P1. The feature enhancement pipeline P1 is one of the dual pipeline branches in the proposed Transformer, and the additional modules involved can be appropriately selected or removed as needed.

template feature vector Φ_{t1}, and its output Φ_{t2} is shown in Eq. (4).

$$\Phi_{t2} = LayerNorm\left(\Phi_{t1} + MultiHead(\Phi_{t1} + p_{t1}, \Phi_{t1} + p_{t1}, \Phi_{t1})\right) \tag{4}$$

where Φ_{t1} also denotes the input of the shared encoder, and p_{t1} represents the corresponding positional encoding.

In Fig. 2, the two multi-head cross-attention submodules, MCA1 and MCA2, have inputs q_{t2} and q_{t3} that are homologous, as they both come from the output Φ_{t2} of the multi-head self-attention submodule. In other words, the multi-head self-attention submodule feeds feature information to both the MCA1 and MCA2 modules. In the MCA1 module, the inputs v_{t2} and k_{t2} are derived from Φ_{sx1}. For the MCA2 module, its input method is similar to that of MCA1, except that the source of v_{t3} and k_{t3} is Φ_{sx3}. Most importantly, the MCA1 and MCA2 modules facilitate the interaction between the feature information extracted from the template and the search region. This interaction serves to enhance the feature information and establish global associations and dependencies between the template and the search region. In detail, due to the attention mechanism, q (q_{t2}, q_{t3}) and k (k_{t2}, k_{t3}) in the MCA1 and MCA2 modules work together to make the target features in v (v_{t2}, v_{t3}) more prominent. Furthermore, the dot product and weighting between q (q_{t2}, q_{t2}) and k (k_{t2}, k_{t3}) are employed to enhance the target features in v (v_{t2}, v_{t3}). This process generates corresponding weights that establish global associations and dependencies between the feature information, thereby accomplishing feature information enhancement. It can be said that this process also involves feature fusion. We can notice that k (k_{t2}, k_{t3}) and v (v_{t2}, v_{t3}) are homologous, which is especially important for the whole global information. It should be noted that the additional modules in the MCA1 and MCA2 modules can be the same or different, which can be determined according to the actual needs. Although the MCA1 and MCA2 modules are identical in structure, they have their own network parameters to obtain rich features. Therefore, the mechanisms of the MCA1 and MCA2 modules can be summarized as Eqs. (5) and (6), respectively.

$$\Phi_{ts1} = LayerNorm\left(\Phi_{t2} + MultiHead(\Phi_{t2} + p_{t2}, \Phi_{sx1} + p_{sx1}, \Phi_{sx1})\right) \tag{5}$$

$$\Phi_{st1} = LayerNorm\left(\Phi_{t2} + MultiHead(\Phi_{t2} + p_{t2}, \Phi_{sx3} + p_{s12}, \Phi_{sx3})\right) \tag{6}$$

where p_{t2}, p_{sx1}, and p_{sx3} are the positional encodings of Φ_{t2}, Φ_{sx1}, and Φ_{sx3}, respectively. It can be observed that Eqs. (5) and (6) are also the two outputs of the shared encoder. Therefore, the shared encoder

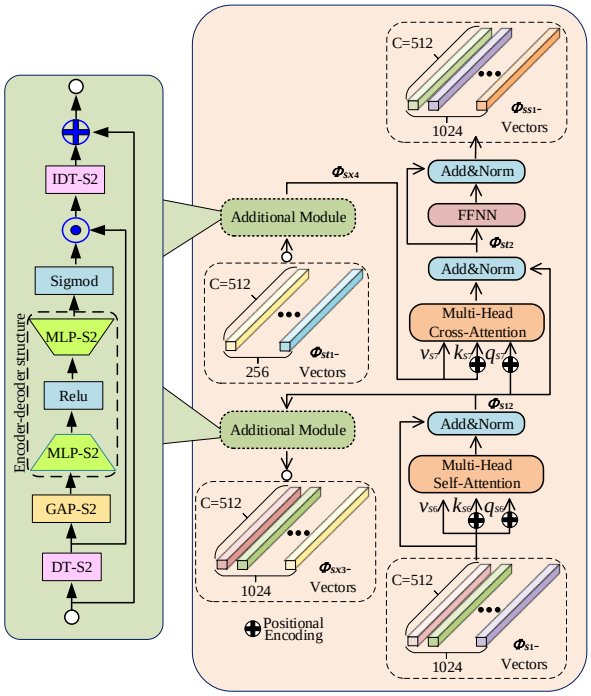

Fig. 4. The network framework of the feature reinforcement pipeline P2. The configurations of Figs. 4 and 3 have similar basic elements and network architectures, which achieve logical functional complementarity, but they do not share network parameters.

can effectively process the template features in the video sequence, which provides a powerful condition for further integrating feature information.

3.2.3. Feature Enhancement Pipeline P1

Figure 3 illustrates the network structure of the feature enhancement pipeline P1, which includes a multi-head self-attention submodule and a multi-head cross-attention submodule with a feedforward neural network. These components are connected to the extended additional modules. Similar to the shared encoder, in the constituent units of the feature enhancement pipeline P1, each element is connected through the residual form, which is particularly beneficial for increasing the depth of the network. Additionally, each submodule still requires a layer normalization operation. For the feedforward neural network, it is mainly used to fit the corresponding feature information, and its output function is $f(\Phi) = max(0, \Phi W_{\Phi 1} + b_{\Phi 1}) W_{\Phi 2} + b_{\Phi 2}$, where Φ, W_j, and $b_j (j = \Phi 1, \Phi 2)$ represent the input, weight parameter matrix, and bias, respectively. Therefore, the mechanism of the feature enhancement pipeline P1 is as follows:

For the multi-head self-attention submodule, the inputs q_{s4}, k_{s4}, and v_{s4} are all derived from the search region feature vector Φ_{s1}, so its output Φ_{s11} can be described as Eq. (7).

$$\Phi_{s11} = LayerNorm\big(\Phi_{s1} + MultiHead(\Phi_{s1} + p_{s1}, \Phi_{s1} + p_{s1}, \Phi_{s1})\big) \tag{7}$$

where p_{s1} stands for the positional encoding. After the multi-head self-attention submodule completes the first processing of the search region features, it realizes the self-enhancement of the target features to a certain extent.

Next, the output Φ_{s11} of the multi-head self-attention submodule flows into the multi-head cross-attention submodule, which serves as the source of inputs k_{s5} and v_{s5}. Meanwhile, the feature vector Φ_{s11} can also be processed by the additional module, and the obtained feature vector Φ_{sx1} can be used as the basis for interaction with the shared encoder. In a sense, the additional modules can act as the connecting nodes between the feature enhancement pipeline P1 and the shared encoder. Moreover, the input q_{s5} of the multi-head cross-attention submodule is derived from the output Φ_{sx2} of the additional module, and Φ_{sx2} already contains the fused feature information from the template and the search region. In this way, according to the attention mechanism, the joint action from q_{s5}, k_{s5} and v_{s5} can realize the association between global

features and achieve the secondary enhancement of feature information. That is to say, the feature vector Φ_{ts2} reflects the performance and association of Φ_{sx2} in the features extracted from the search region, thereby completing the information exchange. It should be noted that the above analysis process is premised on the existence of additional modules, which is of great significance for extending network functions. Therefore, the intermediate output Φ_{ts2} can be written as Eq. (8).

$$\Phi_{ts2} = LayerNorm\big(\Phi_{sx2} + MultiHead(\Phi_{sx2} + p_{sx2}, \Phi_{s11} + p_{s11}, \Phi_{s11})\big) \tag{8}$$

where p_{sx2} and p_{s11} are the corresponding positional encodings, respectively. Besides, when the output function of the additional module connected with q_{s5} is defined as $AM(\cdot)$, the corresponding output can be expressed as $\Phi_{sx2} = AM(\Phi_{ts1})$. However, considering that no additional modules are employed, its output can be described as $\Phi_{sx2} = \Phi_{ts1}$.

Subsequently, the intermediate output Φ_{ts2} will go through a feedforward neural network connected in the residual form, to obtain the output Φ_{tt1} from the multi-head cross-attention submodule. In other words, the mechanism of the feature enhancement pipeline P1 can be summarized as Eq. (9).

$$\Phi_{tt1} = LayerNorm(\Phi_{ts2} + max(0, \Phi_{ts2}W_1 + b_1)\, W_2 + b_2) \tag{9}$$

where W_j and $b_j (j = 1,2)$ represent the corresponding weight parameter matrix and bias, respectively.

3.2.4. Feature Enhancement Pipeline P2

As presented in Fig. 4, the feature enhancement pipeline P2 has a similar network framework to the feature enhancement pipeline P1, which is also composed of a multi-head self-attention submodule, a multi-head cross-attention submodule, and additional modules. For feature enhancement pipelines P2 and P1, although their basic components are the same, they have independent network parameters, which is beneficial for obtaining richer feature information and capturing global dependencies. Furthermore, another particularly distinct difference between feature enhancement pipelines P2 and P1 is that they have different output destinations, and the output uses of the two multi-head self-attention submodules are also different. Specifically, the inputs *q*, *k*, and *v* of the multi-head cross-attention submodules in the two feature enhancement pipelines originate from different feature vectors, respectively. Contrasting feature enhancement pipeline P1, we can find this feature. In the feature enhancement pipeline P2, the output Φ_{s12} of the multi-head self-attention submodule is used as the source of q_{s7}, while k_{s7} and v_{s7} can be traced back to the feature vector Φ_{sx4}. When the additional module is removed, the feature vector Φ_{sx4} degenerates into the output Φ_{st1} of the shared encoder. Meanwhile, k_{s7} and v_{s7} come directly from the output vector Φ_{st1}. Therefore, the intermediate output of the multi-head cross-attention submodule can be expressed as Eq. (10).

$$\Phi_{st2} = LayerNorm\big(\Phi_{s12} + MultiHead(\Phi_{s12} + p_{s12}, \Phi_{sx4} + p_{sx4}, \Phi_{sx4})\big) \tag{10}$$

where p_{st2} and p_{sx4} indicate the corresponding positional encodings. At this point, Φ_{st2} flows through the feedforward neural network, to obtain the output of the feature enhancement pipeline P2, as shown in Eq. (11).

$$\Phi_{ss1} = LayerNorm(\Phi_{st2} + max(0, \Phi_{st2}W_3 + b_3)\, W_4 + b_4) \tag{11}$$

where W_j and $b_j (j = 3,4)$ denote the weight parameter matrix and bias from the feedforward neural network, respectively. For the multi-head self-attention submodule in the feature enhancement pipeline P2, its reasoning principle can refer to the relevant content in the feature enhancement pipeline P1.

Similar to the feature enhancement pipeline P1, in the feature enhancement pipeline P2, the feature vector Φ_{st2} represents the representation and association of the search region features in Φ_{sx4}, which can also be regarded as the result of the joint action of q_{s7} and k_{s7} to enhance the target feature information in v_{s7}. It can be said that, in a certain sense, the feature enhancement pipeline P2 puts more emphasis on the performance of the search region features in the fusion features. However, for the feature enhancement pipeline P1, it highlights the performance of the template features in fused features. Referring to the representations of feature enhancement pipelines P1 and P2, it can be seen that they form a logical complement of functions, to obtain complementary feature information. Therefore, feature enhancement pipes P1 and P2 can also be used as feature fusion units to enhance target features.

3.2.5. Pipeline Feature Fusion Head

The pipeline feature fusion head serves as a hub connecting the feature enhancement pipelines P1 and P2, where the output information from both pipelines converges and interacts. Fig. 5 shows the network architecture of the pipeline feature fusion head, which has the exact opposite logic to the feature enhancement

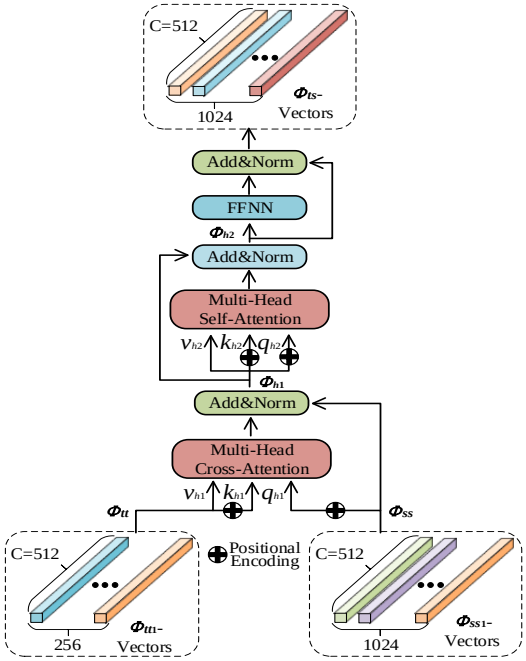

Fig. 5 The network architecture of the pipeline feature fusion head. To some extent, the proposed pipeline feature fusion head also has a feature enhancement effect.

pipelines. It can be seen that the multi-head self-attention submodule and the multi-head cross-attention submodule in the residual form are the main components of the pipeline feature fusion head, which completes the highly fused feature information from the template and the search region. Moreover, its input is the feature output of the feature enhancement pipelines P1 and P2. When the number of modules N in Fig. 1 is equal to 1, the branch output Φ_{tt1} of the feature enhancement pipeline P1 (when N is greater than 1, $\Phi_{tt1} = \Phi_{tt}$) can become the source of k_{h1} and v_{h1} in the input of the pipeline feature fusion head. The branch output Φ_{ss1} of the enhancement pipeline P2 (when N is greater than 1, $\Phi_{ss1} = \Phi_{ss}$) can be used as q_{h1} in the input of the pipeline feature fusion head. Therefore, the corresponding output Φ_{h1} after the multi-head cross-attention submodule completes feature fusion is shown in Eq. (12).

$$\Phi_{h1} = LayerNorm\big(\Phi_{ss1} + MultiHead(\Phi_{ss1} + p_{ss1}, \Phi_{tt1} + p_{tt1}, \Phi_{tt1})\big) \tag{12}$$

where p_{ss1} and p_{tt1} are still the positional encodings of the corresponding feature vectors. When N is greater than 1, the input of the pipeline feature fusion head will change, so it is necessary to replace Φ_{tt1} and Φ_{ss1} in Eq. (12) with Φ_{tt} and Φ_{ss}, respectively, to obtain the refreshed input representation.

Subsequently, the feature vector Φ_{h1} flows through the multi-head self-attention submodule connected to the feedforward neural network, and the intermediate output Φ_{h2} can be obtained, which is written as equation (13).

$$\Phi_{h2} = LayerNorm\big(\Phi_{h1} + MultiHead(\Phi_{h1} + p_{h1}, \Phi_{h1} + p_{h1}, \Phi_{h1})\big) \tag{13}$$

where p_{h1} means the positional encoding. So far, the mechanism of the pipeline feature fusion head can be summarized as Eq. (14).

$$\Phi_{ts} = LayerNorm(\Phi_{h2} + max(0, \Phi_{h2}W_5 + b_5)\,W_6 + b_6) \tag{14}$$

where W_j and $b_j (j = 5,6)$ represent the weight parameter matrix and bias from the feedforward neural network, respectively. The feature vector Φ_{ts} contains abundant enhanced feature information and exhibits significant global feature dependencies, which are all derived from the outstanding feature processing capability of the proposed network.

3.3. Training Loss

The training loss plays a key role in the proposed Repformer. In our method, the tracking prediction network contains two branch structures, namely the classification branch and the regression branch. Therefore, our training loss is a combined loss consisting of a regression loss [37] and a classification loss

[40]. For the regression loss, we use the L1 loss $L_{L1}(\cdot)$ and the gIoU loss $L_{giou}(\cdot)$ [37] to calculate the loss value, so the function composition of the regression loss $L_{reg}(\cdot)$ can be described as Eq. (15).

$$L_{reg} = \sum_{i=1}^{n}(\lambda_{L1}L_{L1}(bbox_i, bbox_i') + \lambda_{giou}L_{giou}(bbox_i, bbox_i')) \quad (15)$$

where $bbox_i$ and $bbox_i'$ are the ground truth and predicted value of the bounding box corresponding to the *i*th training data, respectively, and *n* represents the total number of training data. In addition, λ_{L1} and λ_{giou} denote the hyperparameters of L1 loss and gIoU loss, respectively, which are used to balance the two loss terms. Here, we take $\lambda_{L1} = 4$ and $\lambda_{giou} = 2$.

Meanwhile, the cross-entropy loss [41] is adopted as the classification loss L_{cla} in our training loss, as shown in Eq. (16).

$$L_{cla} = -\sum_{i=1}^{n}(Y_i log P_i + (1 - Y_i)\log(1 - P_i)) \quad (16)$$

where Y_i and P_i denote the label value and corresponding probability value of the *i*th training data, respectively.

Finally, the functional representation of the combined loss can be obtained, which is summarized as Eq. (17).

$$Loss = L_{reg} + \lambda_{cla}L_{cla} \quad (17)$$

where λ_{cla} means the hyperparameter of the regression loss, and λ_{cla} can be set to 5.

4. Experiments

4.1. Experimental Details

The proposed Repformer method is implemented on a specific device with an Intel Core i7-10700k 3.8GHz CPU and Nvidia RTX 2080Ti GPU, which adopts python3.7.10 and pytorch1.5.1 framework. In our method, the number of modules *N* is taken to 2. Several benchmarks, including COCO [41], LaSOT [42] and GOT-10k [43], are used as training sets to train our tracking model offline, and the diversity of training data is increased through data augmentation methods (such as rotation and mirroring) to improve the performance of the tracker. Additionally, the proposed method requires training for 4000 epochs, and each epoch contains 1000 iterations. The AdamW [44] optimizer is employed to update the tracking model, with an initial learning rate of 1e-4. To enhance the training process, the learning rate is not fixed but decays to 0.1 times its original value after 2000 epochs during training.

4.2. Evaluation Metrics

Our experiments are performed on seven tracking benchmarks, including LaSOT [42], GOT-10k [43], UAV123 [45], NfS [46], OTB2015 [47], VOT2018 [48], and TempleColor128 [49]. For LaSOT, UAV123, NfS, OTB2015, and TempleColor128 [42, 49, 44-47], the one-pass evaluation method is utilized to evaluate the proposed tracker. It should be emphasized that the success and precision plots are mainly involved in the one-pass evaluation method. In detail, the success plot refers to the percentage of video frames successfully tracked by the tracker under different overlap thresholds, and the area under the curve (AUC) in the success plot is used to judge the performance of the corresponding tracker. Meanwhile, the precision plot means the percentage of video frames where the distance between the ground truth and the tracking result falls within a given threshold. This measure is also referred to as the precision score. Generally speaking, the AUC score

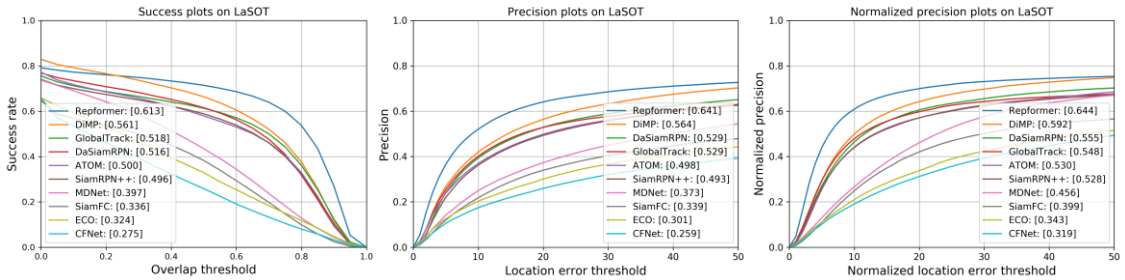

Fig. 6 Evaluation results on LaSOT. The results displayed from left to right represent the success plots, precision plots, and normalized precision plots, with corresponding scores reported in the legends.

Table 1 Comparative results on GOT-10k, involving average overlap (AO), and success rates ($SR_{0.50}$ and $SR_{0.75}$) with overlapping thresholds of 0.50 and 0.75, respectively. Tracker ranking is mainly based on the average overlap score, and the two best-performing metrics are shown in red and blue fonts.

Method	AO(%)	$SR_{0.50}$(%)	$SR_{0.75}$(%)
MDNet[52]	32.5	32.8	9.9
Method in [63]	36.4	37.9	12.6
SiamFC[25]	34.8	35.3	9.8
Ocean[34]	61.1	72.1	47.3
CCOT[20]	32.5	32.8	10.7
ECO[21]	31.6	30.9	11.1
ATOM[33]	55.6	63.4	40.2
KYS[32]	63.6	75.1	51.5
SiamRPN++[29]	51.7	61.6	32.5
D3S[53]	59.7	67.6	46.2
SiamCAR[31]	56.9	67.0	41.5
STMTrack[9]	64.2	73.3	57.5
SiamMCAR[11]	58.4	67.9	44.0
DiMP[51]	61.1	71.7	49.2
TRAT[60]	60.8	72.0	46.7
OUPT[12]	64.6	75.4	54.3
TT-DiMP[62]	64.0	74.4	53.9
DCFST[54]	63.8	75.3	49.8
SiamR-CNN[30]	64.9	72.8	59.7
HCAT[8]	65.3	76.8	57.0
SiamMask[2]	51.4	58.7	36.6
TransT[66]	67.1	76.8	60.9
STARK[67]	68.0	77.7	62.3
Ours	68.5	78.4	63.0

is mainly adopted to evaluate the overall performance of the tracker. For GOT-10k [43], it is required to upload the tracking results to the official website and complete the automatic evaluation according to a specific evaluation protocol. Evaluation metrics contain the average overlap score (AO) between the ground truth and the tracking results, and the success rate score (SR) satisfying the corresponding overlap threshold conditions. When evaluating on VOT2018 [48], the specified evaluation tool is employed to compute the expected average overlap score (EAO), which evaluates the overall performance of our method by combining accuracy and robustness.

4.3. Evaluation and Comparative Experiments

Our tracker is compared with state-of-the-art methods on the aforementioned benchmarks. It can be noticed that these diverse video sequences are captured in real scenes, which are fully qualified for the evaluation needs.

LaSOT. LaSOT [42] is a high-quality large-scale visual tracking benchmark that contains rich challenging attributes in the wild environment. Our Repformer is evaluated on the test set of the LaSOT benchmark [42], which comprises 280 video sequences. Each video sequence has an average length of over 2500 frames. As shown in Fig. 6, our Repformer is compared with some state-of-the-art trackers, including GlobalTrack [50], DiMP [51], SiamRPN++ [29], ECO [21], ATOM [33], MDNet [52], DaSiamRPN [28], CFNet [18], and SiamFC [25]. It can be seen that our tracking method achieves leading performance, achieving an AUC score of 61.3%, a precision score of 64.1%, and a normalized precision score of 64.4%, respectively. In detail, our method outperforms DiMP [51] by 5.2%, 7.7%, and 5.2% in terms of the AUC score, precision score, and normalized precision score, respectively. Compared to the popular GlobalTrack [50] and SiamRPN++ [29], the proposed method achieves 9.5%/11.7% and 11.2%/14.8% gains in the AUC score and precision score, respectively. Additionally, our tracker outperforms other Siamese-based trackers, such as SiamFC, and correlation filtering-based trackers, such as ECO [21]. These comparative results show that in the wild environment, our Repformer has outstanding adaptability to various challenging attributes.

Table 2 Comparison results of AUC scores on UAV123, NfS, and OTB2015. The two best results are shown in red and blue fonts, respectively.

Method	UAV123(%)	NfS(%)	OTB2015(%)
ECO[21]	52.2	46.6	69.1
MDNet[52]	52.8	42.9	67.3
UPDT[22]	54.5	53.7	---
ATOM[33]	64.2	58.4	66.9
Method in [63]	52.2	---	66.5
CRCDCF[7]	51.9	---	---
SiamRPN++[29]	61.3	50.2	69.6
AS2RCF[56]	51.8	---	58.0
SiamMCAR[11]	---	---	66.1
MEGTCF[65]	50.2	---	67.8
SiamRPN++_TGSR[4]	---	52.0	69.8
TT-DiMP[62]	64.3	63.4	69.8
SiamPCF[10]	64.1	59.7	70.0
DiMP[51]	65.3	62.0	68.4
SiamCAN[59]	64.8	---	70.5
HCAT[8]	63.6	63.6	68.1
Ours	66.3	65.2	68.2

Table 3 Comparison results on VOT2018. We present the expected average overlap score (EAO), the accuracy score (Acc), and the robustness score (Rob), respectively. In addition, we use the EAO score to rank the trackers. The two best results are highlighted in red and blue fonts.

Method	EAO(↑)	Acc(↑)	Rob(↓)
ECO[21]	0.280	0.484	0.276
CPT[48]	0.339	0.506	0.239
RCO[48]	0.376	0.507	0.155
SiamRPN[27]	0.384	0.588	0.276
ATOM[33]	0.401	0.590	0.204
LADCF[55]	0.389	0.503	0.159
SiamRPN++[29]	0.414	0.600	0.234
DaSiamRPN[28]	0.383	0.586	0.276
DiMP[51]	0.440	0.597	0.153
OUPT[12]	0.436	0.621	0.170
CRCDCF[7]	0.312	---	---
TrDiMP_PAF[6]	0.458	---	---
SRN[68]	0.242	0.503	---
SiamMask[2]	0.387	0.642	0.295
MEGTCF[65]	0.278	0.505	0.314
SiamRPN++_TGSR[4]	0.440	0.601	0.206
SiamCAN[59]	0.462	0.605	0.183
Ours	0.487	0.632	0.166

GOT-10k. GOT-10k [43] is a large-scale tracking benchmark focused on generic targets, whose test set consists of 180 video sequences. Additionally, it can be noticed that 32 motion patterns and 84 target categories are also included in the test set. Crucially, the test and training sets in GOT-10k do not contain overlapping target categories [43], which helps to improve and evaluate the generalization ability of the model. The comparison results of our tracker with some representative methods are presented in Table 1. Our Repformer obtains an AO score of 68.5%, a $SR_{0.50}$ score of 78.4%, and a $SR_{0.75}$ score of 63.0%, which are 3.2%/1.6%/6.0%, 4.5%/4.0%/9.1%, and 3.6%/5.6%/3.3%, respectively, higher than the state-of-the-art HCAT [8], TT-DiMP [62], and SiamR-CNN [30]. Compared with the popular OUPT [12] and STMTrack [9], our tracker outperforms by 3.9%/3.0%/8.7% and 4.3%/5.1%/5.5% on the AO score, $SR_{0.50}$ score, and $SR_{0.75}$ score, respectively. Moreover, our method outperforms other Siamese-based trackers, such as

Table 4 Comparison results on TempleColor128. Here, performance metrics include the AUC score and accuracy score, and the reported trackers are ranked by the AUC score. The two best indicators are highlighted in red and blue fonts, respectively.

Method	AUC(%)	Precision(%)
SiamFC[25]	50.5	69.4
MEEM[58]	50.0	70.8
CFNet[18]	45.6	60.7
Staple[19]	50.9	66.8
KCF[14]	41.8	58.8
CCOT[20]	58.1	70.4
Self-SDCT[57]	54.0	72.9
CSVMF[64]	48.1	75.2
SRN[68]	56.0	---
CFIT[69]	48.6	66.1
CLIP[70]	53.1	73.2
Method in [61]	52.0	70.4
SiamMCAR[11]	57.2	77.9
SiamCAN[59]	58.0	78.0
SiamATL[71]	57.7	79.4
BAASR[5]	57.8	78.3
ADCF[71]	57.9	78.5
ESiamFC[3]	52.3	66.4
CSTNet[72]	58.5	80.4
CRCDCF[7]	58.4	81.3
Ours	58.9	79.5

SiamMCAR [11], and trackers employing update modules, such as Ocean [34]. Meanwhile, our method also outperforms the remaining trackers in terms of AO scores, such as SiamR-CNN [30], TransT [66], and STARK [67]. Among the state-of-the-art trackers mentioned in Table 1, it can be noticed that the proposed tracker has outstanding generalization ability.

UAV123. The UAV123 benchmark [45] is obtained by a low-altitude UAV and includes 123 video sequences with large differences in target scales. The motion of the camera and tracking targets makes this benchmark more challenging, due to the presence of undesirable effects such as occlusion and fast motion. Thence, this benchmark can evaluate the real-time performance of the tracker in the real environment. Table 2 presents the comparison results of our Repformer with some state-of-the-art real-time tracking methods, including HCAT [8], DiMP [51], SiamPCF [10], TT-DiMP [62], CRCDCF [7], SiamCAN [59], UPDT [22], SiamRPN++ [29], etc. It can be found that our tracker achieves the best AUC score of 66.3%, outperforming the popular DiMP [51] and HCAT [8] trackers by 1% and 2.7%, respectively. Therefore, our tracker has a protruding advantage in real-time scenarios.

NfS. The NfS benchmark [46] consists of 100 video sequences with high-speed moving targets in real scenes, including 30fps and 240fps versions captured by the high-speed camera. The proposed method is evaluated on the 30fps version of the NfS benchmark [46], and the obtained AUC scores are shown in Table 2. From Table 2, when compared with other state-of-the-art trackers, our method achieves the most prominent AUC score of 65.2%, which is 1.6% higher than the second-ranked tracker. Compared with some representative trackers, such as TT-DiMP [62], SiamPCF [10], and DiMP [51], the proposed method outperforms them by 1.8%, 5.5%, and 3.2%, respectively, which shows that our method is promising in practical scenarios.

OTB2015. OTB2015 [47] is a typical tracker evaluation benchmark consisting of 100 video sequences with 11 challenging attributes. It can be seen from Table 2 that the proposed method achieves an AUC score of 68.2%. Although our method's AUC score on OTB2015 [45] may not be the brightest, the performance obtained is still competitive with other representative tracking methods.

VOT2018. The VOT2018 benchmark [48] contains 60 fully-annotated video sequences with various perturbing attributes, such as illumination, occlusion, and scale variation. We also evaluate the proposed

Table 5 The running speed of the proposed Repformer on several typical benchmarks. We rely on the number of video sequence frames per second processed by the tracking method.

Benchmark	LaSOT[42]	GOT-10k[43]	UAV123[45]	NfS[46]	OTB2015[47]
Speed(fps)	55.7	50.9	59.6	57.6	62.7

Table 6 The running speed of the proposed Repformer and some state-of-the-art methods. The two best results are represented in red and blue fonts, respectively.

Method	Speed(fps)	Method	Speed(fps)
STMTrack[9]	37.0	Ocean[34]	52.0
ATOM[33]	28.0	SiamCAR[31]	52.3
DiMP[51]	29.0	SiamMCAR[11]	28.0
TRAT[60]	28.0	SiamRPN++[29]	53.0
TT-DiMP[62]	21.4	SiamCAN[59]	45.0
TransT[66]	50.0	STARK[67]	41.8
MEGTCF[65]	11.3	CLIP[70]	35.0
CRCDCF[7]	5.0	SiamMask[2]	55.0
SRN[68]	26.1	Ours	57.3

Table 7 Comparative results of ablation experiments on LaSOT. Each serial number represents the corresponding network structure, where √ means the corresponding element exists, and × means that the corresponding element is deleted. Additionally, AM represents the additional module. The best results are shown in red fonts.

Serial Number	SE			FEP-P1		FEP-P2		PFFH	AM	AUC (%)	Pr (%)	NPr (%)
	SE-S	SE-C1	SE-C2	P1-S	P1-C	P2-S	P2-C					
NO.1	×	√	√	√	√	√	√	√	√	60.1	62.9	63.3
NO.2	√	×	√	√	√	√	√	√	√	59.3	62.1	62.4
NO.3	√	√	×	√	√	√	√	√	√	59.1	62.5	62.9
NO.4	√	√	√	×	√	√	√	√	√	60.0	62.8	63.2
NO.5	√	√	√	√	×	√	√	√	√	56.7	59.4	59.8
NO.6	√	√	√	√	√	×	√	√	√	59.9	62.3	62.5
NO.7	√	√	√	√	√	√	×	√	√	57.8	60.6	61.1
NO.8	√	√	√	√	√	√	√	×	√	55.6	58.7	59.0
NO.9	×	×	×	×	×	×	×	√	√	30.5	33.4	33.8
NO.10	√	√	√	√	√	√	√	√	×	50.2	53.4	54.1
NO.11	√	√	√	√	√	√	√	√	√	61.3	64.1	64.4

Repformer on VOT2018 [48] and compare it with the state-of-the-art tracking methods, involving SiamCAN [59], CRCDCF [7], OUPT [12], SiamRPN [27], DiMP [51], LADCF [55], DaSiamRPN [28], ATOM [33], SiamRPN++ [29], MEGTCF [65], SiamMask [2], SiamRPN++_TGSR [4], etc. In general, the EAO score is adopted to evaluate the overall performance of the tracking method, and Table 3 presents the evaluation results. It can be noted that our tracker achieves the best EAO score of 48.7%, which means that our method achieves the most outstanding overall performance. When compared with SiamCAN [59], the proposed tracker obtains a gain of 2.5% for the EAO score. Meanwhile, our method also achieves state-of-the-art performance with a score of 63.2% in terms of accuracy, outperforming the OUPT [12] and SiamRPN++ [29] trackers by 1.1% and 3.2%, respectively. Furthermore, our method also achieves relatively favorable results in terms of the robustness score (ranked fourth), ranking fourth with a mere 0.6% difference from the third-ranked LADCF [55].

TempleColor128. As shown in Table 4, we compare the proposed tracker with typical tracking methods on TempleColor128 [49], including CRCDCF [7], SiamCAN [59], SiamMCAR [11], CCOT [20], Self-SDCT [57], KCF [14], CFNet [18], Staple [19], SRN [68], CFIT [69], CLIP [70], SiamATL [71], CSTNet [72], ADCF [71], ESiamFC [3], BAASR [5], MEEM [58], and SiamFC [25]. It can be observed from Table

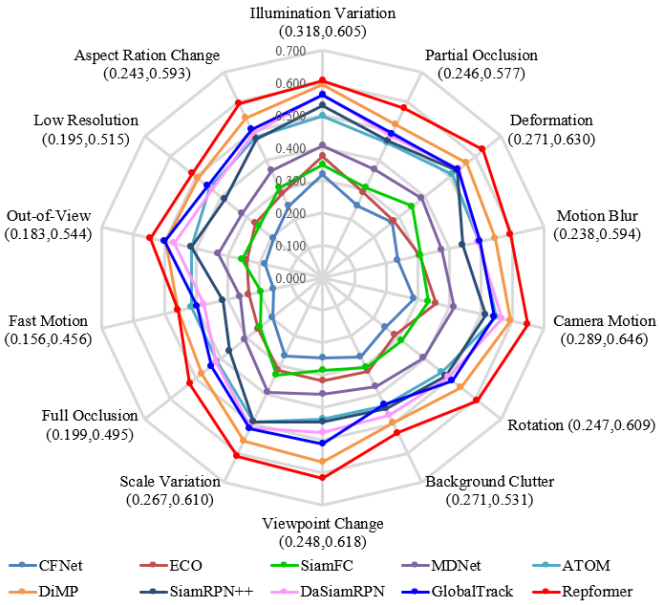

Fig. 7 Comparison of AUC scores for 14 challenging attributes on the LaSOT test set.

4 that our tracker achieves the highest AUC score of 58.9%, while achieving the second highest precision score of 79.5%. Therefore, our Repformer still achieves impressive performance on TempleColor128 [49].

Running Speed Analysis. When conducting the evaluation experiments described above, the proposed Repformer is chosen to perform running speed statistics and analysis on LaSOT [42], GOT-10k [43], UAV123 [45], NfS [46], and OTB2015 benchmarks [47]. Table 5 presents the corresponding statistical results. From Table 5, it can be seen that our Repformer achieves the highest running speed of 62.7fps on OTB2015 [47], while the lowest speed is also 50.9fps. In other words, our tracking method runs at more than 50.0fps on all 5 evaluation benchmarks, which can satisfy real-time performance.

Our Repformer also implements a speed comparison with some state-of-the-art tracking methods, such as STMTrack [9], ATOM [33], TT-DiMP [62], SiamCAN [59], DiMP [51], CRCDCF [7], SiamMCAR [11], Ocean [34], SiamCAR [31], and SiamRPN++ [29], as presented in Table 6. It should be noted that the running speed of our method takes the average speed (57.3fps) in Table 5 as the comparison standard. From Table 6, it can be seen that the proposed Repformer achieves the best running speed among many popular tracking methods. Besides, SiamMask[2], ranked second, also performs well with a running speed of 55.0fps, which is 2.3fps slower than our method. When compared with Ocean [34], SiamCAR [31], and SiamCAN [59], our method outperforms them by 5.3fps, 5.0fps, and 12.3fps, respectively. For other trackers, such as TT-DiMP [62], STARK [67], CRCDCF [7], and SiamMCAR [11], the proposed Repformer exhibits a particularly notable advantage in running speed. Therefore, our method can also meet the requirements for real-time tracking.

From the above evaluation process, it can be seen that the proposed Repformer performs well in tracking accuracy and running speed.

4.4. Ablation Experiments

In this section, the related ablation experiments are conducted on LaSOT [42] to verify the role of each fundamental element in the proposed method. We retrain the proposed Repformer while evaluating it on the test set. Table 7 presents the comparative results of the ablation experiments. Since the additional modules are used for the reference of extended functions, additional modules are not considered in the ablation experiments, so the corresponding results are not reflected in Table 7. Moreover, the evaluation metrics included in Table 7 involve the AUC score, precision score, and normalized precision score (the last three columns in the table), and we adopt the AUC score to analyze the overall performance of the corresponding tracker. It should be noted that we use our Repformer as a baseline method and regard its results as a standard reference, as shown in NO.11.

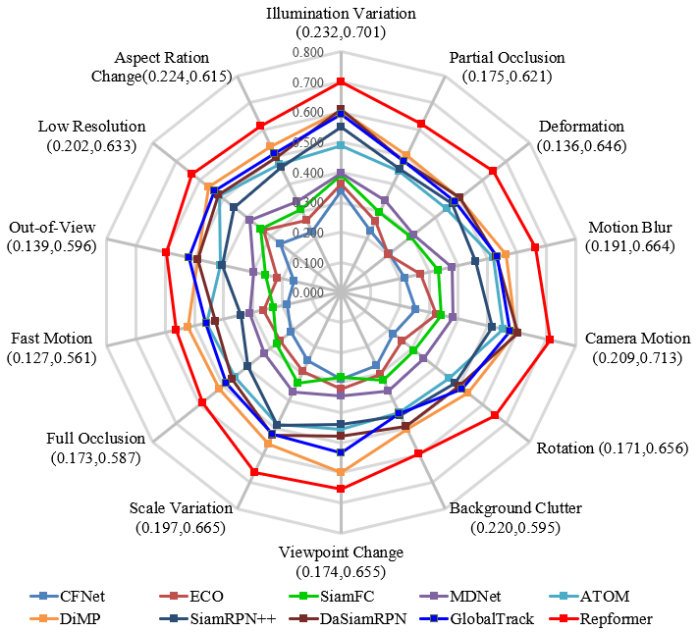

Fig. 8 Comparison of precision scores for 14 challenging attributes on the LaSOT test set.

In this experiment, we specifically investigate the roles of the components within the shared encoder, the feature enhancement pipelines (P1, P2), and the pipeline feature fusion head. For simplicity, we denote the shared encoder as SE, while the feature enhancement pipelines P1 and P2 are denoted by FEP-P1 and FEP-P2, respectively. In SE, the multi-head self-attention submodule can be replaced by SE-S, and the two multi-head cross-attention submodules can be denoted as SE-C1 and SE-C2, respectively. For FEP-P1 and FEP-P2, the corresponding submodules can be expressed as P1-S/ P1-C and P2-S/ P2-C respectively. As shown in NO.1 from Table 7, compared with the baseline method NO.11, the AUC score drops by 1.2%, which reflects the importance of SE-S for the template feature processing. For NO.2 and NO.3, when SE-C1 and SE-C2 are removed, the corresponding performance metrics drop by 2.0% and 2.2%, respectively, which indicates that SE-C1 and SE-C2 are valuable for the feature interaction between the template and the search region. It can be noticed that the AUC scores between NO.2 and NO.3 differ by 0.2%, which means that the effects of NO.2 and NO.3 are particularly close to a certain extent. Meanwhile, when NO.1 is compared with NO.2 and NO.3, it can be observed that SE-C1 and SE-C2 play a more positive role in the overall performance of the tracker. When P1-S and P1-C in FEP-P1 are removed, NO.4 and NO.5 present the AUC scores of 60.0% and 56.7%, respectively. It can be seen that the influence of P1-C on the overall performance is more prominent than that of P1-S. For NO.6 and NO.7, their overall performance is similar to that of NO.4 and NO.5. Furthermore, in FEP-P1 and FEP-P2, both P1-S and P2-S exhibit similar effects on the overall performance, as do P1-C and P2-C. When replacing PFFH with a correlation operation, we obtain an AUC score of 55.6% under the corresponding tracker, which shows that PFFH is more advantageous in integrating feature information. As can be seen from NO.9, the AUC score of our method drops to 30.5% when only PFFH is retained in the proposed method. Compared with NO.8, the whole of SE, FEP-P1, and FEP-P2 has a more profound impact on the overall performance of the tracker. When NO.10 is compared with NO.11, the AUC score corresponding to NO.10 decreases by 11.1%, indicating that the additional module has a positive effect on improving the performance of the tracker. As shown in NO.11, in our comprehensive ablation experiments, the proposed baseline method achieves state-of-the-art performance, which demonstrates that our method is of great significance for the efficient utilization of feature information.

4.5 Attribute-based evaluation

To comprehensively evaluate the performance of the proposed Repformer across various complex and challenging attributes, we conduct attribute-based evaluation experiments on the LaSOT test set with some typical trackers, including SiamRPN ++ [29], MDNet [52], DaSiamRPN [28], DiMP [51], ATOM [33], GlobalTrack [50], ECO [21], SiamFC [25], and CFNet [18]. In addition, the 14 challenging attributes

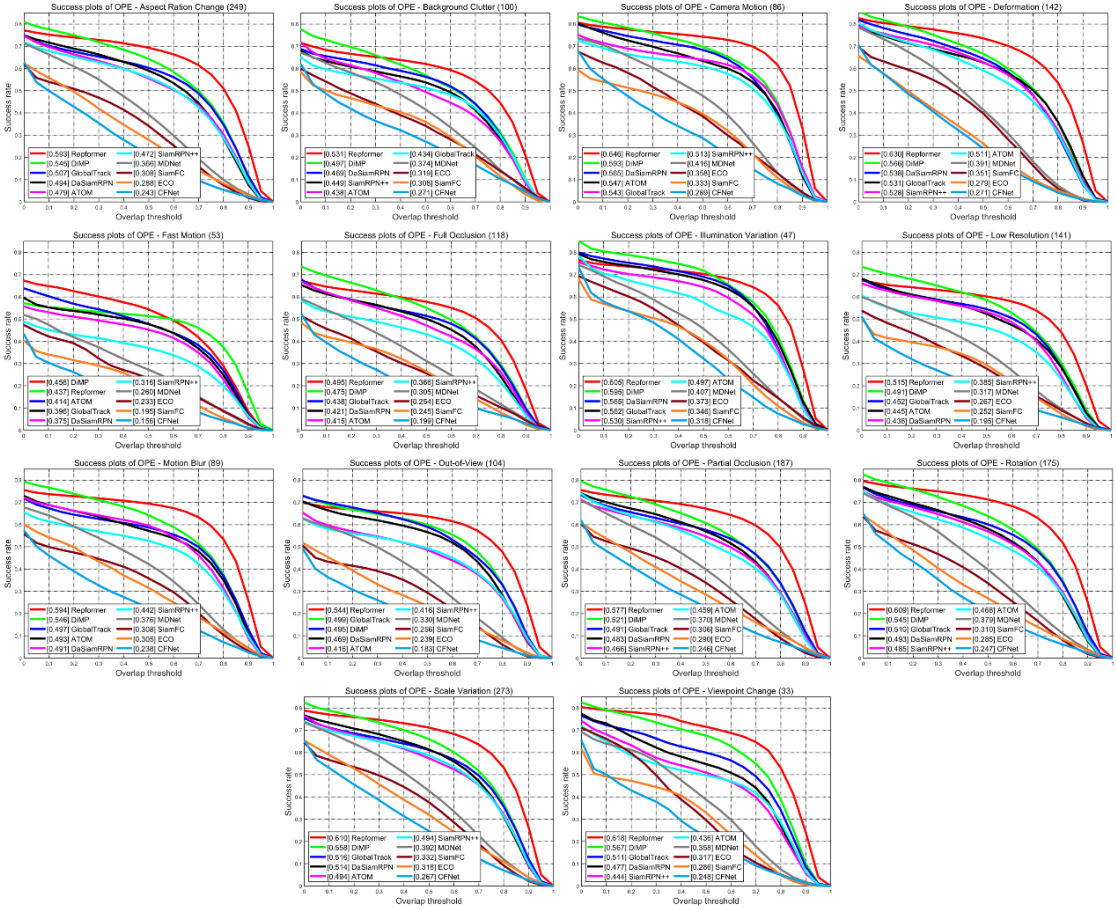

Fig. 9 Success plots for 14 challenging attributes on the LaSOT test set.

involved include scale variation, rotation, illumination variation, deformation, motion blur, etc. Figs. 7 and 8 show the comparison results of the AUC and precision scores, respectively. From a qualitative point of view, it can be noted in Figs. 7 and 8 that the score curve of our method always surrounds the remaining tracking methods, which shows that our method has achieved outstanding performance in all 14 challenging attributes. Therefore, our method also obtains the best overall performance.

From Figs. 9 and 10, the success and precision plots for the reported tracker can be observed, respectively. From a quantitative perspective, it is possible to see the scores of all trackers on different challenging attributes. For motion blur, our tracker is 4.8% higher than DiMP [51] on the AUC score. When considering scale variation, the proposed method outperforms DaSiamRPN [28] by 9.6% in terms of the AUC score, while obtaining a precision score of 63.6%. In terms of deformation, our tracker also achieves an AUC score of 63.0%, which is superior to other methods. For the remaining challenging attributes, the proposed method still achieves outstanding tracking performance.

4.6. Qualitative Analysis and Visualization

It is a particularly challenging task to achieve robust visual tracking in complex scenes. To further illustrate the effectiveness of the proposed Repformer, qualitative analysis and corresponding visualizations are performed during tracking, to name a few. Here, some common targets are chosen to complete the experiment, such as various ships and marine organisms. Specifically, our method is qualitatively compared with state-of-the-art methods on 4 challenging video sequences, including GlobalTrack [50], SiamRPN++ [29], and ECO [21], as shown in Fig. 11. Additionally, it can be noticed that the selected video sequences involve large ships (t_{11}~t_{14}), small speedboats (t_{31}~t_{34}), turtles (t_{21}~t_{24}), and sharks (t_{41}~t_{44}).

In practical environments, it is inevitable to encounter adverse distractions such as motion blur, scale

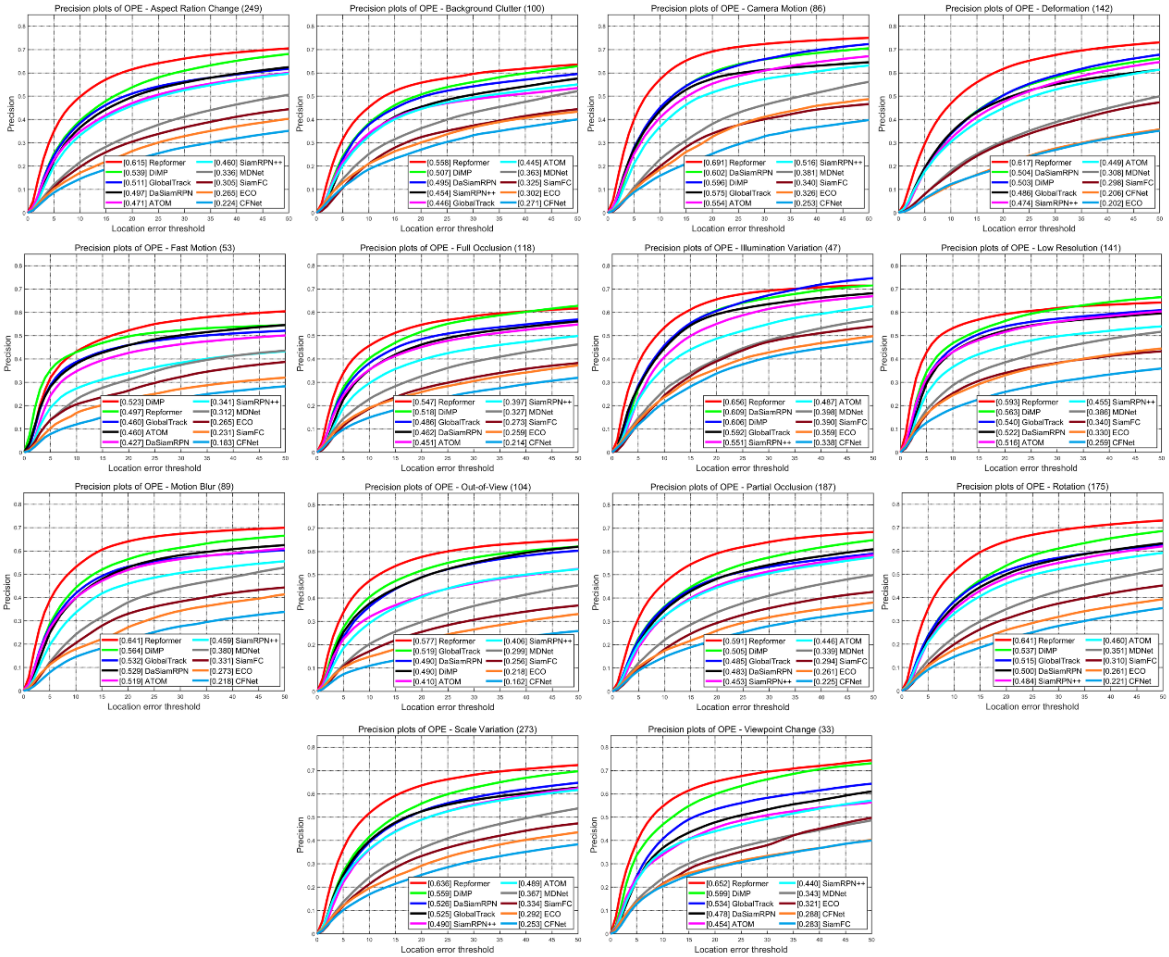

Fig. 10 Precision plots for 14 challenging attributes on the LaSOT test set.

variation, and illumination. These undesirable effects are also widespread in our chosen tracking sequences, which are captured from complex scenes. For large ships (t_{11}~t_{14}), our Repformer can represent the bounding box of the ship in real-time and achieve accurate tracking, as shown in Fig. 11. However, other tracking methods only track a certain part of the ship body or suffer from tracking drift, such as ECO [21] and GlobalTrack [50]. This illustrates that our method can efficiently utilize the feature information of the tracking target, while avoiding taking a part of the ship body as a whole and overcoming the interference of relative motion and fast motion between the camera and the target. When tracking small speedboats, all trackers face unfavorable conditions of out-of-view, fast motion, scale variation, etc. Specifically, in t_{31}, it can be found that most trackers have a large degree of bounding box shift, such as ECO [21], but the tracking bounding box of our method remains near the ground truth in the field of view (the same is true in t_{32} ~ t_{34}). Compared with t_{31}, the small speedboat in t_{32} ~ t_{34} encounters obvious rotation, fast motion, seawater fluctuation, and scale variation. It can be observed that each tracker achieves excellent tracking results in t_{33}. In addition, especially in t_{34} with blur, our tracker still stands out. Compared with tracking ships on the water, it is also particularly difficult to track underwater marine organisms. From t_{21}~t_{24}, when tracking the target turtle, some interference factors inevitably appear, including attitude change, scale variation, and cluttered background. It can be observed that no matter how the turtle changes its swimming posture, the tracking methods involved can give its position well. More importantly, our proposed method can consistently and robustly present bounding boxes around the ground truth, which is difficult for ECO [21], as shown in t_{21}~t_{24}. For t_{41}~t_{44}, the target shark exhibits flexibility in its movement, leading to dramatic changes in appearance and posture. Moreover, illumination, motion blur, sea swell, and the presence of other fish are also some interference factors that cannot be ignored. When the trackers GlobalTrack [50], SiamRPN++ [29], and ECO

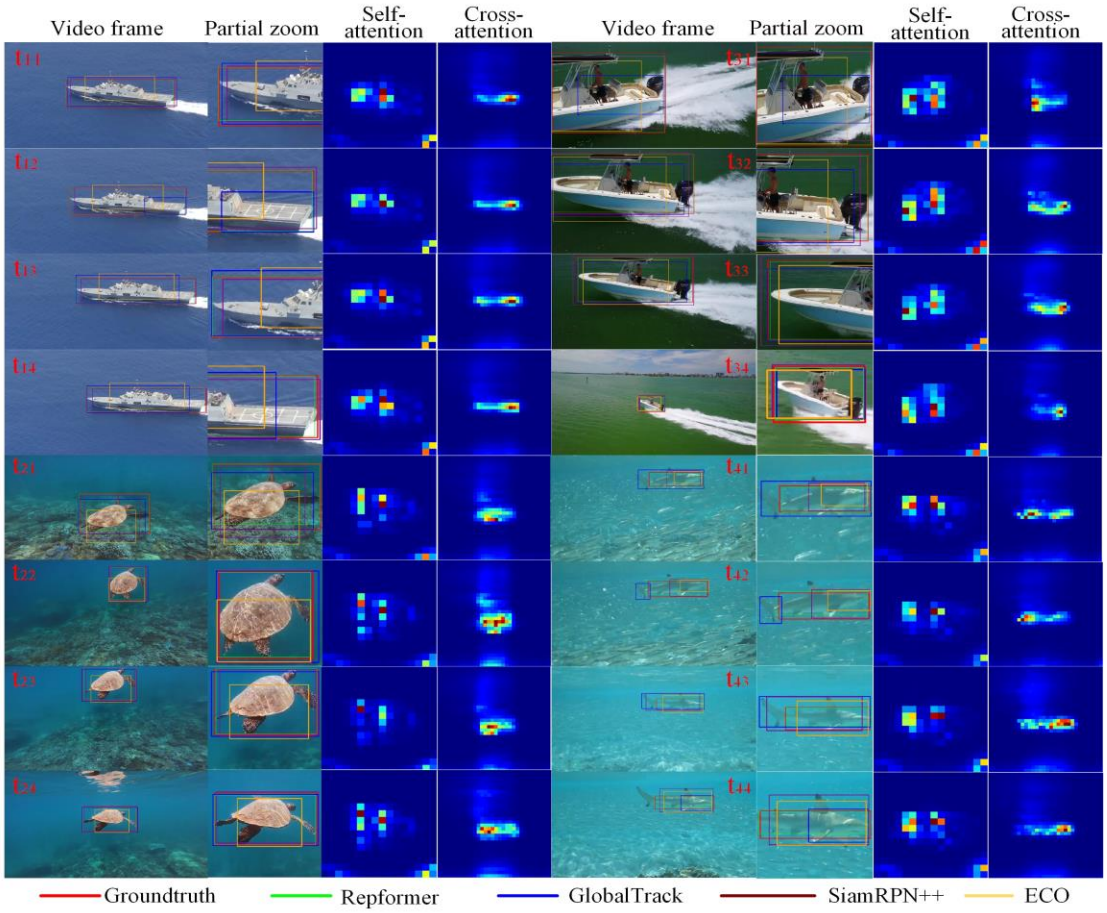

Fig. 11 Visualization of tracking results from our Repformer and state-of-the-art methods. To present specific details, we perform local amplification while showing some instances of attention maps in the proposed network.

[21] track the target shark, robust tracking cannot be guaranteed continuously. Especially in t_{42}, GlobalTrack [50] experiences an undesirable tracking drift. For SiamRPN++ [29] and ECO [21], their tracking results are also far from the ground truth in the same environment, such as t_{41} and t_{42}. However, the tracking effect achieved by our tracking method is still the closest to the ground truth. It can be noted that numerous disadvantages are encountered when tracking target sharks, which is an extremely serious challenge for any tracking method.

Fig. 11 also lists some attention maps from the self-attention and cross-attention mechanisms in the proposed network. It can be observed that in the selected 4 video sequences, the distribution of self-attention maps is relatively scattered, while the distribution of cross-attention maps is relatively concentrated. Besides, the self-attention maps contained in each video sequence have similar distributions within groups, except that they highlight different key points, as do the cross-attention maps. It can be noticed that the self-attention mechanism attempts to identify all target features within the field of view, placing a particular emphasis on capturing key feature information. For the cross-attention mechanism, to locate the range of the tracking target, the cross-attention mechanism tends to more important feature information. From the root, the cross-attention mechanism has effectively fused features derived from the template and search region, while playing the role of reinforcing features to a certain extent.

Therefore, our Repformer achieves outstanding tracking performance, thanks to the excellent integration ability of the proposed method for the feature information from the template and search region.

5. Discussion

During the training phase, the adopted benchmarks cover a variety of complex scenarios. In the evaluation

phase, to verify the tracking performance of the proposed Repformer, evaluation experiments are performed on 7 popular benchmarks, including quantitative analysis and qualitative analysis. In the quantitative analysis, we conduct a comparative study in terms of tracking accuracy and speed to examine the quantitative metrics of our tracker. For qualitative analysis, 4 challenging video sequences are selected to complete the evaluation and present the visualization results. Globally, our method gains outstanding performance compared with representative state-of-the-art methods. Specifically, our method achieves the best performance metrics on LaSOT [42], GOT-10k [43], UAV123 [45], NfS [46], VOT2018 [48], and TempleColor128 [49], while also obtaining a relatively favorable AUC score on OTB2015 [47]. For a variety of complex challenging attributes, the proposed tracker also obtains excellent performance. In addition, Fig. 11 demonstrates that the proposed method still exhibits superiority in complex scenarios involving ships and marine organisms. It can be noticed that both the utilized evaluation benchmarks and the selected video sequences contain various adverse factors, such as illumination variation, occlusion, and scale variation. However, our method has clear advantages when compared with representative methods. It cannot be ignored that Table 6 presents that our tracker also achieves a competitive running speed. These experimental results indicate that our Repformer can achieve excellent tracking performance in complex scenarios while possessing outstanding generalization ability, robustness, and real-time performance.

6. Conclusion

In this paper, we propose a novel Siamese-based Repformer tracker with a shared-encoder dual-pipeline Transformer variant to achieve robust visual tracking in complex scenes. The proposed Transformer variant network is composed of the shared encoder, the feature enhancement pipelines (P1, P2), and the pipeline feature fusion head, which employs a hybrid attention mechanism. The developed shared encoder and feature enhancement pipelines (P1, P2) constitute the main unit of our Transformer variant, while the pipeline feature fusion head serves as the final interaction hub responsible for highly integrating the feature information derived from the feature enhancement pipelines. By incorporating a hybrid attention mechanism, our method efficiently utilizes the feature information from the template and search region, thereby establishing dependencies, correlations, and enhancements of global feature information. The proposed tracker also obtains outstanding performance on 14 challenging attributes. Moreover, our method can gain efficient tracking representation through a concise tracking prediction network structure. Extensive experiments demonstrate that our Repformer achieves remarkable performance on multiple tracking benchmarks with a real-time running speed of 57.3fps.

Although the proposed method has achieved outstanding performance, it also involves certain limitations. Our tracker faces difficulties in handling the situations when the tracking target is completely obscured by distractions or moves out of view, primarily due to the absence of a re-detection mechanism in the tracking framework. Furthermore, our tracking framework is sequential, and the interactivity between features can be further enhanced. Therefore, future work can include the following aspects: 1) incorporating a re-detection mechanism into the overall tracking network to regain target features, which can effectively deal with the cases where the target is completely occluded or moves out of view; 2) designing a hierarchical network structure to improve the feature interaction capability of the tracking network, which enables more efficient feature integration.

Acknowledgment

We are very grateful to the editors and anonymous reviewers for their constructive comments and suggestions to improve our manuscript. Moreover, this work is supported by the Natural Science Foundation of Heilongjiang Province of China under Grant No. F201123, the National Natural Science Foundation of China under Grant 52171332 and 52075530, the Green Intelligent Inland Ship Innovation Programme under Grant MC-202002-C01, and the Development Project of Ship Situational Intelligent Awareness System under Grant MC-201920-X01.

Statements and Declarations

CRediT Authorship Contribution Statement. Conceptualization: Fengwei Gu, Jun Lu, Chengtao Cai, Qidan Zhu, and Zhaojie Ju; Methodology: Fengwei Gu and Jun Lu; Formal analysis and investigation:

Fengwei Gu, Jun Lu, Chengtao Cai, Qidan Zhu, and Zhaojie Ju; Writing - original draft preparation: Fengwei Gu; Writing - review and editing: Fengwei Gu, Jun Lu, Chengtao Cai, Qidan Zhu, and Zhaojie Ju; Funding acquisition: Jun Lu, Chengtao Cai, Qidan Zhu, and Zhaojie Ju; Resources: Jun Lu, Chengtao Cai, Qidan Zhu, and Zhaojie Ju; Supervision: Jun Lu and Zhaojie Ju.

Data Availability Statement. All data generated or analysed during this study are included in this published article.

Declaration of Competing Interest. The authors have no relevant financial or non-financial interests to disclose.

References

[1] Xu L, Gao M, Liu Z, et al. Accelerated duality-aware correlation filters for visual tracking[J]. Neural Computing and Applications, 2022: 1-16.

[2] Hu W, Wang Q, Zhang L, et al. Siammask: A framework for fast online object tracking and segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(3): 3072-3089.

[3] Huang H, Liu G, Zhang Y, et al. Ensemble siamese networks for object tracking[J]. Neural Computing and Applications, 2022, 34(10): 8173-8191.

[4] Wang H, Liu J, Su Y, et al. Trajectory Guided Robust Visual Object Tracking with Selective Remedy[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2023.

[5] Zhang J, Yuan T, He Y, et al. A background-aware correlation filter with adaptive saliency-aware regularization for visual tracking[J]. Neural Computing and Applications, 2022: 1-18.

[6] Li S, Zhao S, Cheng B, et al. Part-Aware Framework for Robust Object Tracking[J]. IEEE Transactions on Image Processing, 2023, 32: 750-763.

[7] Zhu X F, Wu X J, Xu T, et al. Complementary discriminative correlation filters based on collaborative representation for visual object tracking[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 31(2): 557-568.

[8] Chen X, Wang D, Li D, et al. Efficient Visual Tracking via Hierarchical Cross-Attention Transformer[J]. arXiv preprint arXiv:2203.13537, 2022.

[9] Fu Z, Liu Q, Fu Z, et al. Stmtrack: Template-free visual tracking with space-time memory networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 13774-13783.

[10] Zeng Y, Zeng B, Yin X, et al. SiamPCF: siamese point regression with coarse-fine classification network for visual tracking[J]. Applied Intelligence, 2022, 52(5): 4973-4986.

[11] Yu J, Zuo M, Dong L, et al. The multi-level classification and regression network for visual tracking via residual channel attention[J]. Digital Signal Processing, 2022, 120: 103269.

[12] He X, Chen C Y C. Learning object-uncertainty policy for visual tracking[J]. Information Sciences, 2022, 582: 60-72.

[13] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters[C]//2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010: 2544-2550.

[14] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 37(3): 583-596.

[15] Henriques J F, Caseiro R, Martins P, et al. Exploiting the circulant structure of tracking-by-detection with kernels[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2012: 702-715.

[16] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[17] Li Y, Zhu J. A scale adaptive kernel correlation filter tracker with feature integration[C]//European conference on computer vision. Springer, Cham, 2014: 254-265.

[18] Valmadre J, Bertinetto L, Henriques J, et al. End-to-end representation learning for correlation filter based tracking, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2805-2813.

[19] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P. H. Torr, Staple: Complementary learners for real-time tracking, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1401-1409.

[20] M. Danelljan, A. Robinson, F. S. Khan, M. Felsberg, Beyond correlation filters: Learning continuous convolution operators for visual tracking, in: Proc. European Conference on Computer Vision. Springer, Cham, 2016, pp. 472-488.

[21] M. Danelljan, G. Bhat, K. F. Shahbaz , M. Felsberg, Eco: Efficient convolution operators for tracking, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6638-6646.

[22] G. Bhat, J. Johnander, M. Danelljan, F.S. Khan, M Felsberg, Unveiling the power of deep tracking, in: Proc. European Conference on Computer Vision (ECCV), 2018, pp. 483-498.

[23] Danelljan M, Hager G, Shahbaz Khan F, et al. Learning spatially regularized correlation filters for visual tracking[C]//Proceedings of the IEEE international conference on computer vision. 2015: 4310-4318.

[24] Danelljan M, Hager G, Shahbaz Khan F, et al. Convolutional features for correlation filter based visual tracking[C]//Proceedings of the IEEE international conference on computer vision workshops. 2015: 58-66.

[25] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional siamese networks for object tracking[C]//European conference on computer vision. Springer, Cham, 2016: 850-865.

[26] Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., & Wang, S. (2017). Learning dynamic siamese network for visual object tracking. In Proceedings of the IEEE international conference on computer vision (pp. 1763-1771).

[27] Li, B., Yan, J., Wu, W., Zhu, Z., & Hu, X. (2018). High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8971-8980).

[28] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, W. Hu, Distractor-aware siamese networks for visual object tracking, in: Proc. European Conference on Computer Vision, 2018, pp. 101-117.
[29] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan, Siamrpn++: Evolution of siamese visual tracking with very deep networks, in: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4282-4291.
[30] P. Voigtlaender, J. Luiten, P. H. Torr, B. Leibe, Siam r-cnn: Visual tracking by re-detection, in: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6578-6588.
[31] Guo D, Wang J, Cui Y, et al. SiamCAR: Siamese fully convolutional classification and regression for visual tracking[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 6269-6277.
[32] G. Bhat, M. Danelljan, L. V. Gool, R. Timofte, Know your surroundings: Exploiting scene information for object tracking, in: Proc.European Conference on Computer Vision. Springer, Cham, 2020, pp. 205-221.
[33] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, Atom: Accurate tracking by overlap maximization, in: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4660-4669.
[34] Z. Zhang, H. Peng, J. Fu, B. Li, W. Hu, Ocean: Object-aware anchor-free tracking, in: Proc. European Conference on Computer Vision, 2020 pp. 771-787.
[35] Q. Wang, C. Yuan, J. Wang, W. Zeng, Learning attentional recurrent neural network for visual tracking, IEEE Transactions on Multimedia. 21.4 (2018) 930-942.
[36] Gu F, Lu J, Cai C. RPformer: A Robust Parallel Transformer for Visual Tracking in Complex Scenes[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 1-14.
[37] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Proc. European Conference on Computer Vision. Springer, Cham, 2020, pp. 213-229.
[38] D. Liu, G. Liu, A transformer-based variational autoencoder for sentence generation, in: Proc. 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp.1-7.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
[40] X. Ding, E.C. Larson, Incorporating uncertainties in student response modeling by loss function regularization, Neurocomputing. 409 (2020), 74-82.
[41] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla´r, C. L. Zitnick, Microsoft coco: Common objects in context, in: Proc. European Conference on Computer Vision. Springer, Cham, 2014, pp. 740-755.
[42] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. J. Yu, H.X. Bai, Y. Xu, C. Y. Liao, H.B. Ling, Lasot: A high-quality benchmark for large-scale single object tracking, in: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5374-5383.
[43] L. Huang, X. Zhao, K. Huang, GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild, IEEE Transactions on Pattern Analysis and Machine Intelligence. 43, (2021) 1562-1577.
[44] Loshchilov I, Hutter F. Decoupled weight decay regularization[J]. arXiv preprint arXiv:1711.05101, 2017.
[45] M. Mueller, N. Smith, B. Ghanem, A benchmark and simulator for uav tracking, in: Proc. European Conference on Computer Vision. Springer, Cham, 2016, pp. 445-461.
[46] K. H. Galoogahi, A. Fagg, C. Huang, D. Ramanan, S. Lucey, Need for speed: A benchmark for higher frame rate object tracking, in: Proc. IEEE International Conference on Computer Vision, 2017, pp. 1125-1134.
[47] Y. Wu, J. Lim, M. Yang, Object Tracking Benchmark, in IEEE Transactions on Pattern Analysis and Machine Intelligence. 37 (2015) 1834-1848
[48] M Kristan, et al. The sixth visual object tracking vot2018 challenge results, in: Proc. European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 0-0.
[49] P. Liang, E. Blasch, H. Ling, Encoding color information for visual tracking: Algorithms and benchmark, IEEE Transactions on Image Processing. 24.12 (2015), pp. 5630-5644.
[50] Huang L, Zhao X, Huang K. Globaltrack: A simple and strong baseline for long-term tracking[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07): 11037-11044.
[51] G. Bhat, M. Danelljan, L. V. Gool, R Timofte, Learning discriminative model prediction for tracking, in: Proc. IEEE/CVF International Conference on Computer Vision, 2019, pp. 6182-6191.
[52] H. Nam, B Han, Learning multi-domain convolutional neural networks for visual tracking, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4293-4302.
[53] A. Lukezic, J. Matas, M. Kristan, D3S-A discriminative single shot segmentation tracker, in: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7133-7142.
[54] L. Zheng, M. Tang, Y. Chen, J. Wang, H. Lu, Learning feature embeddings for discriminant model based tracking, in: Proc. European Conference on Computer Vision (ECCV), 23.28, 2020, pp. 759-775.
[55] T. Xu, Z. H. Feng, X. J. Wu, J. Kittler, Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking, IEEE Transactions on Image Processing, (2019), pp.5596-5609.
[56] Zhang J, He Y, Wang S. Learning Adaptive Sparse Spatially-Regularized Correlation Filters for Visual Tracking[J]. IEEE Signal Processing Letters, 2023.
[57] D. Yuan, X. Chang, P. Y. Huang, Q. Liu, Z. He, Self-supervised deep correlation tracking, IEEE Transactions on Image Processing. 30 (2020) 976-985.
[58] J. Zhang, S. Ma, S. Sclaroff, MEEM: robust tracking via multiple experts using entropy minimization, in: Proc. European Conference on Computer Vision. Springer, Cham, 2014, pp. 188-203.
[59] Zhou W, Wen L, Zhang L, et al. SiamCAN: real-time visual tracking based on Siamese center-aware network[J]. IEEE Transactions on Image Processing, 2021, 30: 3597-3609.
[60] Saribas H, Cevikalp H, Köpüklü O, et al. TRAT: Tracking by attention using spatio-temporal features[J]. Neurocomputing, 2022, 492: 150-161.
[61] Yan Y, Guo X, Tang J, et al. Learning spatio-temporal correlation filter for visual tracking[J]. Neurocomputing, 2021, 436: 273-282.
[62] Nie J, Wu H, He Z, et al. Spreading Fine-grained Prior Knowledge for Accurate Tracking[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022.

[63] Zhang H, Cheng L, Zhang T, et al. Target-distractor Aware Deep Tracking with Discriminative Enhancement Learning Loss[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022.
[64] Elayaperumal D, Joo Y H. Robust visual object tracking using context-based spatial variation via multi-feature fusion[J]. Information Sciences, 2021, 577: 467-482.
[65] Ma S, Zhao Z, Hou Z, et al. Correlation Filters Based on Multi-Expert and Game Theory for Visual Object Tracking[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 1-14.
[66] Chen X, Yan B, Zhu J, et al. Transformer tracking[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 8126-8135.
[67] Yan B, Peng H, Fu J, et al. Learning spatio-temporal transformer for visual tracking[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 10448-10457.
[68] Fan N, Liu Q, Li X, et al. Siamese Residual Network for Efficient Visual Tracking[J]. Information Sciences, 2023.
[69] Hu Q, Guo Y, Lin Z, et al. Object tracking using multiple features and adaptive model updating[J]. IEEE Transactions on Instrumentation and Measurement, 2017, 66(11): 2882-2897.
[70] Liu H, Hu Q, Li B, et al. Robust long-term tracking via instance-specific proposals[J]. IEEE Transactions on Instrumentation and Measurement, 2019, 69(4): 950-962.
[71] Huang B, Xu T, Shen Z, et al. SiamATL: online update of siamese tracking network via attentional transfer learning[J]. IEEE Transactions on Cybernetics, 2021.
[72] Yao S, Zhang H, Ren W, et al. Robust online tracking via contrastive Spatio-Temporal aware network[J]. IEEE Transactions on Image Processing, 2021, 30: 1989-2002.
[73] Yuan D, Chang X, Li Z, et al. Learning adaptive spatial-temporal context-aware correlation filters for UAV tracking[J]. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2022, 18(3): 1-18.
[74] Yuan D, Chang X, Liu Q, et al. Active learning for deep visual tracking[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023.
[75] Yuan D, Shu X, Liu Q, et al. Robust thermal infrared tracking via an adaptively multi-feature fusion model[J]. Neural Computing and Applications, 2023, 35(4): 3423-3434.
[76] Yang K, He Z, Pei W, et al. SiamCorners: Siamese Corner Networks for Visual Tracking [J]. IEEE Transactions on Multimedia, 2021, 24: 1956-1967.