symmetry MDPI

Article

SiamRDT: An Object Tracking Algorithm Based on a Reliable Dynamic Template

Qian Zhang, Zihao Wang * and Hong Liang

College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266555, China; zhangqian8266@163.com (Q.Z.); liangh@upc.edu.cn (H.L.)
* Correspondence: z20070077@s.upc.edu.cn

Abstract: Most trackers are only dependent on the first frame as a template to search for and locate the target location in subsequent videos. However, objects may undergo occlusions and deformation over time, and the original snapshot of the object can no longer accurately reflect the current appearance of the object, which greatly limits the performance improvement of the tracker. In this paper, we propose a novel Siamese tracking algorithm with symmetric structure called SiamRDT, which reflects the latest appearance and motion states of objects through additional reliable dynamic templates. The model decides whether to update the dynamic template according to the quality estimation score and employs the attention mechanism to enhance the reliability of the dynamic template, adopting the depth-wise correlation algorithm to integrate the initial template and the dynamic template and the search area. Through reliable dynamic templates and credible initial templates, the model can fuse initial-state information and the latest-state information of objects. We conduct sufficient ablation experiments to illustrate the effectiveness of the proposed key components, and the tracker achieves very competitive results on four large-scale tracking benchmarks, namely OTB100, GOT-10k, LaSOT, and TrackingNet. Our tracker achieves an AO score of 61.3 on GOT-10k, a precision score of 56.5 on LaSOT, a precision score of 69.3 on TrackingNet, and a precision score of 90.5 on OTB100.

Keywords: visual tracking; Siamese network; deep learning; computer vision

check for updates

Citation: Zhang, Q.; Wang, Z.; Liang, H. SiamRDT: An Object Tracking Algorithm Based on a Reliable Dynamic Template. *Symmetry* **2022**, *14*, 762. https://doi.org/10.3390/sym14040762

Academic Editor: Mihai Postolache

Received: 8 March 2022
Accepted: 1 April 2022
Published: 7 April 2022

1. Introduction

Object tracking is a fundamental vision task. It aims to infer the location of an arbitrary target in a video sequence, given only its location in the first frame. The main challenge of tracking lies in that the objects may undergo heavy occlusions, large deformation, and illumination variations [1,2]. Tracking at real-time speeds has a variety of applications, such as surveillance, robotics, autonomous driving, and human-computer interaction [3–5].

For most of the popular trackers (such as SiamFC++ [6], SiamRPN++ [7], Ocean [8], and TransT [9]), the first frame of the objects plays a decisive role in positioning the objects. In other words, most trackers are only dependent on the first frame as a template to search for and locate the target location in subsequent videos. However, objects may undergo heavy occlusions and deformation over time, like Figure 1, and the original snapshot of the object can no longer accurately reflect the current appearance of the object, which limits the tracker to capture the similarities between the initial template and the current video image.

Both spatial and temporal information is important for object tracking. Therefore, some trackers (such as Stark [10–12]) naturally add a template online update mechanism, which continuously provides the status change information of objects across frames and powerfully assists the task of classifying and locating objects. However, this also buries a hidden danger, in that the updated template is predicted by the model rather than the ground truth. If the updated template itself is not accurately located or cannot effectively express object category information, it may mislead the tracker to capture real objects. More importantly, the updated template contains not only foreground object information but also background interference information because of problems such as poor positioning and illumination variations. This naturally introduces an interesting question: is there a way to extract excellent quality dynamic updated template?

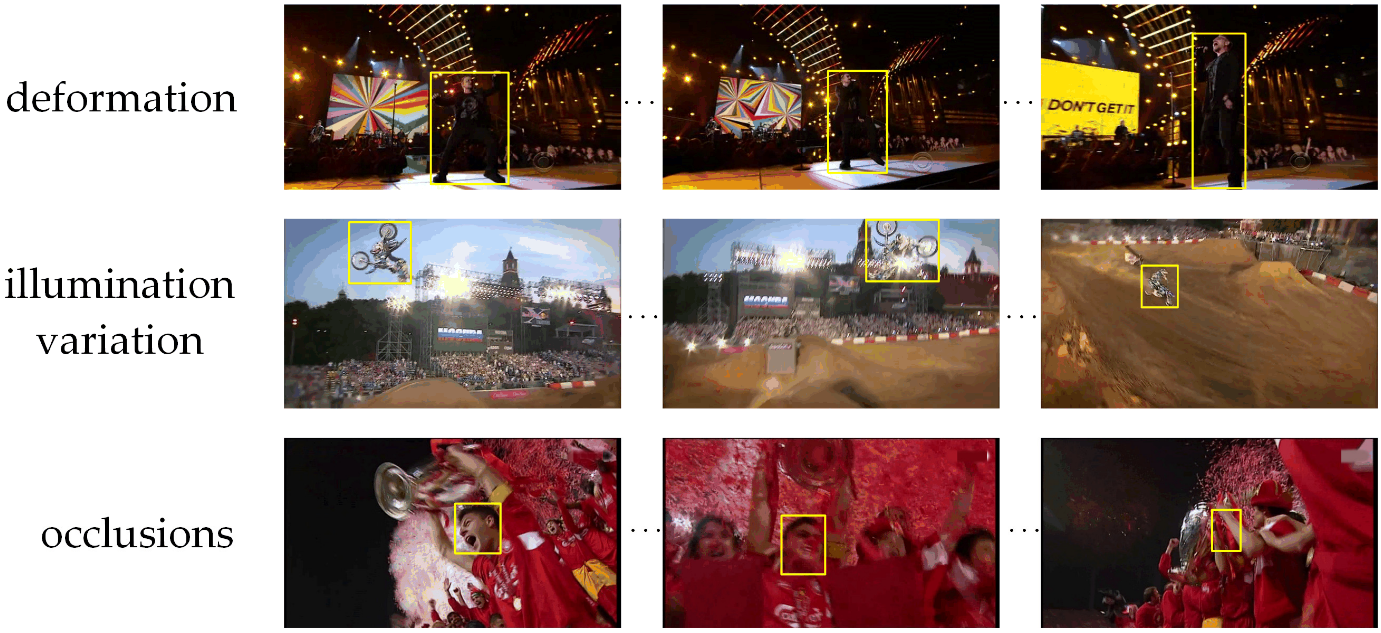

Figure 1. Objects may undergo heavy occlusions, deformation, and rapid illumination variation over time.

In this work, we employ a quality estimation head to estimate the IoU score between the bounding box and the ground truth for updating the dynamic template with better positioning quality. At the same time, inspired by Transformer [13], we resort to the transformer to integrate the initial target information and the dynamically updated information, generating discriminative features, which have strengthened foreground targets and weakened background information. These two parts form a pipeline for updating and enhancing dynamic templates, introducing additional object information.

As shown in Figure 2, The entire tracking algorithm we proposed contains five major components: the feature extraction part, feature enhancement part, feature fusion part, prediction part, and dynamic template updating part. The feature extraction part accepts the initial target object, the current image, and the dynamic update template as input. For better performance, we use deeper ResNet instead of Alexnet as the backbone network of feature extraction. The feature enhancement part enhances the features of the dynamic template and the current search area through the cross-attention mechanism. The feature fusion part has two key roles. One is to fuse the dynamic template and the current search area to extract the deformation information of the target in the time dimension. The other is to fuse the initial template and the current search area to extract the category and boundary information of the target in the spatial dimension. The prediction part includes a classification head to distinguish the foreground and background, a localization head for predicting the boundary box of the tracking target, and a quality score head for judging the accuracy of boundary prediction. The pipeline used to update the dynamic template uses the output of the quality score head to judge whether to update the dynamic template.

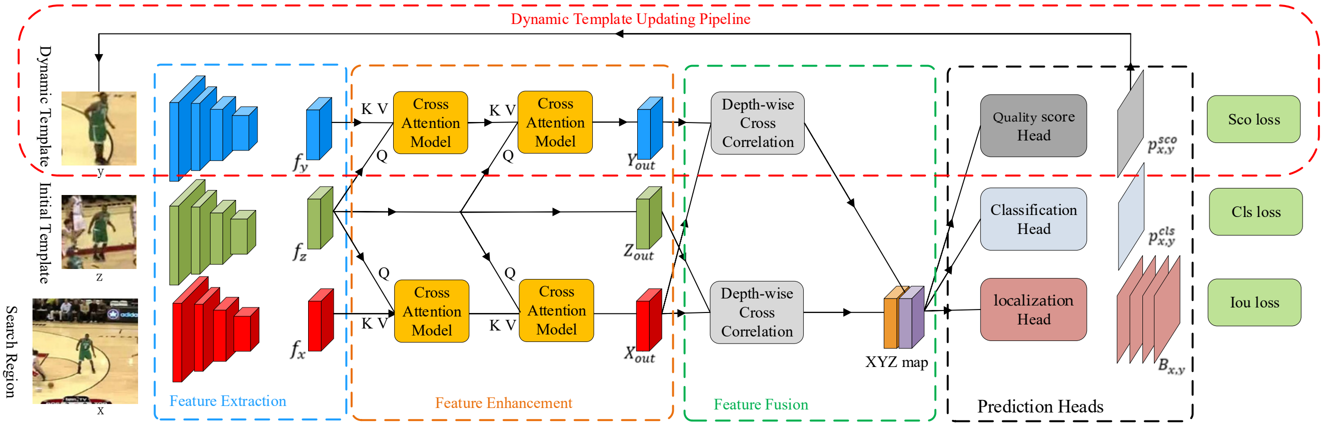

Figure 2. The architecture of SiamRDT. The blue dotted line box is the feature extraction part, the orange dotted line box is the feature enhancement part, the green dotted line box is the feature fusion part, and the black dotted line box is the prediction part. The red dotted box at the top is the whole dynamic template updating part, which flows laterally through the above four parts.

To summarize, the main threefold contributions of this work are listed below:

- We propose a new pipeline for updating and enhancing dynamic templates. It can predict the reliability of the current object state to decide whether to update the dynamic template. Due to the introduction of Transformer, it can also refer to the initial template to adaptively enhance the region of interest in the dynamic template, which further improves the credibility of the dynamic template. Experiments demonstrate that using the cross-attention mechanism to enhance the feature representation of dynamic templates can improve the performance of the tracker.
- We propose a new tracker with a symmetric structure, which can accept input from three aspects, dynamic template, initial template, and current image, and generate output in three aspects: quality score, classification, and localization. It can combine both spatial and temporal information in video sequences.
- Our method achieves favorable performance compared with the state-of-the-art on four tracking challenge benchmarks and can run at real-time speed in GPU. In addition, our proposed dynamic template pipeline can be flexibly applied to other trackers.

2. Related Work

In this section, we summarize the related work on updating the object template, anchor-free mechanism, and transformer architecture in the tracking framework, as well as briefly review recent Siamese trackers.

2.1. Siamese Trackers

In recent years, more and more trackers have adopted the Siamese network architecture [6–9,13–16]. The pioneering work, SiamFC [14], combines feature correlation with Siamese network to offline train a similarity metric between the object target and the current image. However, before SiamRPN++ [7], Siamese trackers usually used shallow backbone networks such as Alexnet [17] as feature extractors. Since SiamRPN++ breaks the spatial invariance restriction of the Siamese tracker, deep feature extraction networks such as Resnet [18] began to show brilliant performance superiority in the tracking field. Following the high-performance universal object tracker design guidelines proposed by SiamFC++, some further improvements such as decomposition of classification and state estimation, estimation quality assessment, and non-ambiguous scoring have been widely adopted. Our tracker performs three sub-tasks: classification, localization, and quality assessment. In the field of object tracking, it is effective to divide classification, localization, and quality assessment into three independent and related subtasks. The classification task aims to determine whether a tracked object exists in a certain search area. The tracker relies on the classification head to separate the foreground from the background, but the classification task cannot accurately delineate the boundaries of objects. Localization tasks aim to capture accurate boundaries of objects that are constantly moving and deforming. High-quality localization results require dynamic templates to provide up-to-date state information for objects. Because dynamic templates need to be constantly updated, we add a separate quality estimation task to control template updates. Experiments show that using quality estimation scores to control template updates improves performance compared to classification confidence.

2.2. Anchor-Free Tracking Mechanism

In recent years, the straightforward and effective anchor-free method [19–21] in the field of object detection has gradually replaced the complex region proposal method and has become mainstream. In addition, in the field of single object tracking, some trackers have adopted anchor-free mechanisms and have achieved success. By directly predicting the center point offset, width, and height, or predicting the top-left and the bottom-right corners, the prediction and training time is greatly saved. Matching between objects and anchors turns out to generate a false positive result, causing the tracking to fail. Instead of adjusting the anchor box, we directly predict the corners of sub-windows corresponding to the pixels in the target area.

2.3. Updating the Object Template

Trackers based on correlation operations and Siamese networks have been widely studied and applied. They usually have clean architecture, higher frame rates, and better tracking performance, but in the case of deformation or occlusion or illumination variation or scale variation or fast motion, trackers lacking online update modules have difficulty classifying objects and locating objects. In our work, we analyze the defects of existing template update modules and design a novel pipeline for template update.

Accurately locating the object is a complex task. Accurate estimation of the bounding box relies heavily on the state information of the object's posture, shape, and angle of view. It requires a wealth of prior knowledge about the target to be tracked. However, the single target tracking task only holds the initial information of the object in the first frame. To adapt to the object variations, some trackers have introduced a template update strategy. Most of the sample and templates are updated at regular intervals. This update mechanism is unreliable in some tracking situations. The updated template may not be accurately positioned, and a large area of the object may be obscured. These reasons will cause the updated template to not reflect the latest state change information of the object. SiamRCR adds the quality evaluation branch to assist classification, and we use the output of the quality score branch to judge whether to update the dynamic template. We only use the latest dynamic template with the high score, which contains the latest shape and position information of the current target. Too many historical templates will increase the algorithm overhead and reduce the running speed.

2.4. Transformer in Tracking

Transformer is an architecture for machine translation proposed by Vaswani et al. Transformer establishes the dependency relationship between word sequences through self-attention and strengthens the expression of the corresponding word embedding according to the degree of association. By establishing the relationship between all word embedding, Transformer can understand the global information of the text sequence. The traditional convolution method increases the receptive field by stacking convolution kernels, but it is still difficult to cover the global information.

Inspired by the successful application of Transformer to natural language processing tasks, many works combining Transformer have emerged in the field of computer vision, such as ViT [22], DETR [23], and TransT [9]. ViT divides the two-dimensional image into patches, and each image patch is flattened into a one-dimensional vector. Then the pure

Transformer architecture can be used to directly model the relationship between image patches. For the first time, DETR uses Transformer in the field of target detection, using encoders, decoders, and CNN to establish the connection between the object and the global context. TransT introduces the attention mechanism into the field of target tracking. The self-attention is used to enhance the effective features of the current image and the initial template image. The cross-attention is employed to establish the connection between the two. In this work, inspired by cross-attention, we use the initial template as the query of the cross-attention module and use the dynamic template as the key and value to enhance the valuable foreground object information in the dynamic template, thereby enhancing the credibility of the dynamic template.

3. Proposed Methods

In this section, we introduce the proposed Siamese tracker for visual tracking, called SiamRDT. As shown in Figure 2, the network architecture consists of five major components: feature extraction part, feature enhancement part, feature fusion part, prediction part, and dynamic template updating part. It operates as follows. The tracker accepts three inputs: current frame, initial template, and dynamic template. First, they are all fed into the feature extraction backbone network to generate their corresponding features. Then, the features of the initial template and the dynamic template are input into the cross-attention module to directionally enhance the target feature expression in the dynamic template. In the same way, the features of the initial template and the current image are input into the cross-attention module to enhance the target information in the search area in advance. Then, in the feature fusion module, the enhanced features are fused through deep cross-correlation operation, the relevant feature map is created, and it is further input into the corresponding quality evaluation, classification, and localization head. The quality assessment score determines whether to update the dynamic template with the current picture. The key components are shown in detail as follows.

3.1. Feature Extraction Part

We utilize the modified ResNet-50 as the backbone network of our SiamRDT. Choosing the deeper network can extract deeper abstract semantic features. The performance of the Siamese network-based tracking algorithm can be significantly boosted if it is armed with much deeper networks, but the original ResNet-50 was designed for classification tasks. It has a large stride of 32 pixels, which means that the generated feature maps have fuzzy location information, which is not conducive to localization tasks. So, we reduce the effective strides from 32 pixels to 8 pixels by removing the last stage of ResNet50 and modifying the fourth stage to have a unit spatial stride. Besides, we utilize dilated convolution [24] in the fourth stage for the higher receptive field. To reduce the number of subsequent parameters, we added a 1 × 1 convolutional layer at the end to reduce the channel dimension of the feature map from 1024 to 256. Our tracker accepts three inputs: an initial template image $z \in \mathbb{R}^{3 \times H_z \times W_z}$, a dynamic template image $y \in \mathbb{R}^{3 \times H_y \times W_y}$, and an image of the current search area $x \in \mathbb{R}^{3 \times H_x \times W_x}$. After extracting features through Resnet with an equivalent stride of 8 pixels, the corresponding feature maps are $f_z \in \mathbb{R}^{256 \times \frac{H_z}{8} \times \frac{W_z}{8}}$, $f_y \in \mathbb{R}^{256 \times \frac{H_y}{8} \times \frac{W_y}{8}}$, $f_x \in \mathbb{R}^{256 \times \frac{H_x}{8} \times \frac{W_x}{8}}$.

3.2. Feature Enhancement Part

In this section, we take the enhanced dynamic template as an example to introduce the feature enhancement part. The dynamic template features f_y extracted by the backbone network need to perform cross-attention enhancement operations with the initial template features f_z to highlight the tracking object features. As shown in Figure 3, our cross-attention augmentation module consists of multi-head dot-product attention, feed-forward networks, residual connection, layer normalization, and positional encoding. Unlike the self-attention in [25], our residual connection part connects the key with the output of the multi-head attention part because our key and query come from different matrices.

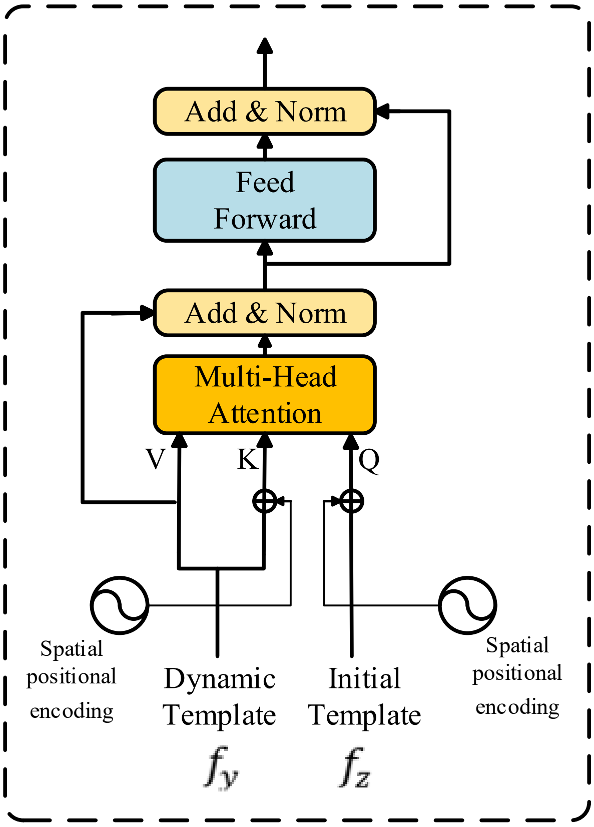

Figure 3. The architecture of cross-attention modules.

3.2.1. Multi-Head Dot-Product Attention

We regard both f_y and f_z as a vector group with $\frac{H}{8} \times \frac{W}{8}$ 256-dimensional vectors and connect the feature vectors into a sequence as the input of the scaled dot-product attention function, like Figure 4, which is defined as:

$$Att(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where the query Q means the initial template sequence, the key K and value V mean the dynamic template sequence, and d_k is the dimension of the key.

As stated in [25], multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. We project the query key value into different subspaces for attention-weighted enhancement, then concatenate and project the individual subspaces results back to the original dimension, resulting in the final values. The multi-head attention is defined as:

$$H_i = Att\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2}$$

$$MutiHead(Q, K, V) = Concat(H_1, H_2, \ldots, H_n)W^O \tag{3}$$

where the projection matrix W_i^Q, $W_i^K \in \mathbb{R}^{d_m \times d_k}$, $W_i^V \in \mathbb{R}^{d_m \times d_v}$, and $W^O \in \mathbb{R}^{nd_v \times d_m}$. In our work, we employ n = 8 parallel attention heads, and $d_m = 256$, $d_k = d_v = \frac{d_m}{n} = 32$. The input dimension is the same as the output dimension.

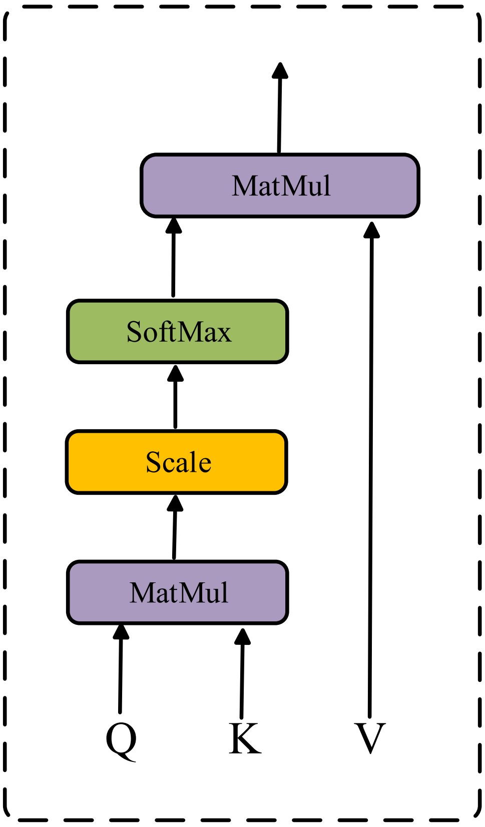

Figure 4. The architecture of dot-product attention.

3.2.2. Feed-Forward Network

We add a fully connected feed-forward network after the attention layer to enhance the fitting ability of the model. It consists of two linear transformations with a ReLU activation function in the middle.

$$FFN(X) = max(0,\ XW_1 + b_1)W_2 + b_2 \tag{4}$$

where X means the output of the attention layer, W_1, W_2 means two different transformation matrices, and b_1, b_2 are two bias vectors.

The module overview is as follows:

$$Y_{att} = f_y + MutiHead\big(f_z + PE_{sin},\ f_y + PE_{sin}, f_y\big) \tag{5}$$

$$Y_{out} = Y_{att} + FFN(Y_{att}) \tag{6}$$

where $f_z,\ f_y \in \mathbb{R}^{256\times(\frac{H}{8}\times\frac{W}{8})}$ mean the extracted feature maps, PE_{sin} is the positional encoding using the sin function, Y_{att} is the output of the attention layer, and Y_{out} is the output after the feed-forward neural network.

3.2.3. Summary of Feature Enhancement Part

In fact, we stacked two cross-attention modules. As shown in Figure 5, Y_{out} must be fed into the cross-attention module again as K, V to get the final output. This section describes how to use the f_z to enhance the f_y to obtain the Y_{out}. We use the same method to get X_{out} from f_z and f_x. X_{out} is the enhanced feature matrix of the current search area.

Send f_y, f_z, and f_x into the feature enhancement part, and the obtained feature matrix is Y_{out}, Z_{out}, and X_{out}. Because the feature enhancement part retains the original features of the initial template, so $Z_{out} = f_z$.

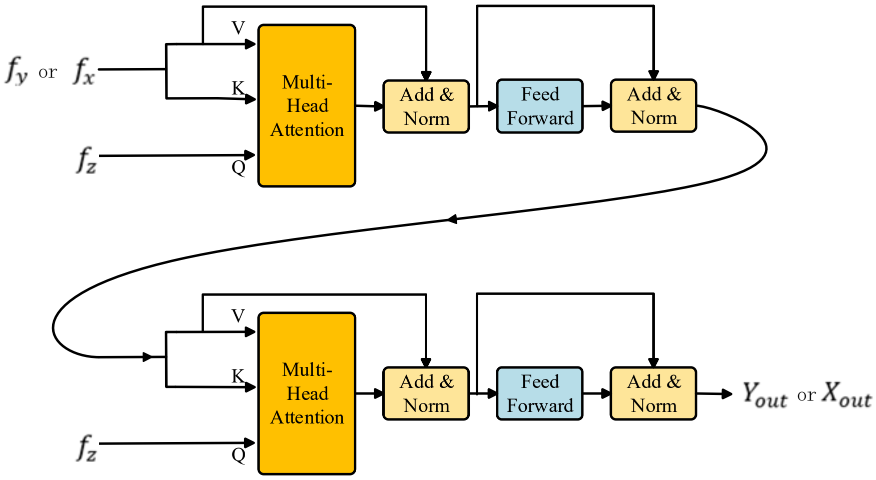

Figure 5. The architecture of the feature enhancement part.

3.3. Feature Fusion Part

This work uses depth-wise cross correlation to fuse the feature matrix of the dynamic template, initial template, and search area, like in Figure 6. In the tracking algorithm, the cross-correlation algorithm is generally used to calculate the response map. The response map shows the matching similarity of the template in different regions of the current image. SiamFC uses the cross-correlation algorithm to calculate the result as a single channel response map. The single-channel response map loses the rich features of different channels, so this work adopts depth-wise cross correlation, which can capture multi-channel correlation features between templates and search patches. In depth-wise cross correlation, two feature maps with the same number of channels are correlated channel by channel. X_{out}, Z_{out}, and Y_{out} have the same number of channels: 256 each. As shown in Figure 6, we perform depth-wise cross correlation operations on X_{out} using Y_{out} and Z_{out}, respectively. Then we concat the response matrix of the two branches. Traditional cross correlation relies on stacking convolution kernels to increase the number of channels in the response map. The advantage of adopting depth-wise correlation is that it consumes less computing cost and memory to obtain a 512-channel response map. Through the correlation operations at the channel level, the response map contains the semantic information of different channels. By merging the two branches, the response map contains the spatiotemporal information of the objects.

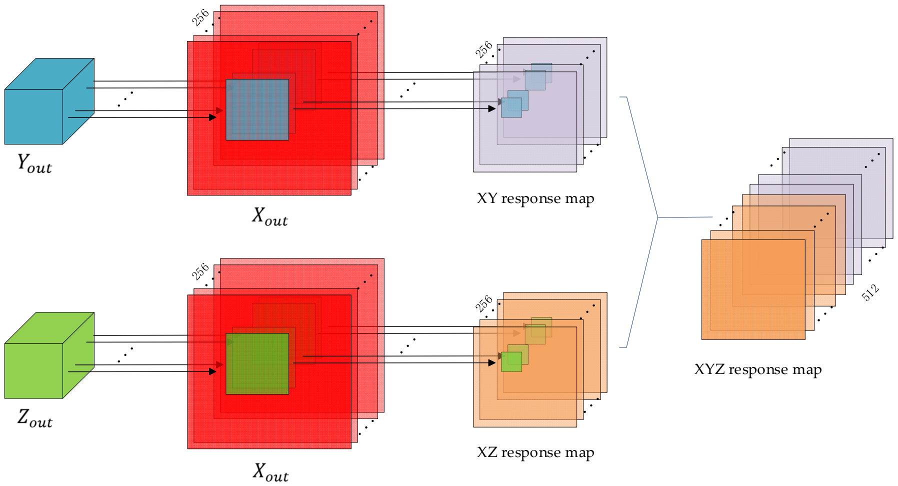

Figure 6. The architecture of the feature fusion part.

3.4. Prediction Part

In our work, we adopt three prediction heads for three subtasks: quality estimation head, classification head, and localization head. Our prediction head consists of a three-layer perceptron of hidden dimension d with activation functions between the layers. Each location (x, y) on the feature map output by the prediction head corresponds to an image patch centered at location $\left(\frac{s}{2} + sx, \frac{s}{2} + sy\right)$ on the input image, and s means the backbone strides of 8 pixels.

The quality score estimation head is used to evaluate the IOU score of the bounding box predicted by the localization head for the corresponding coordinate position, and to determine whether to update the dynamic template. Because [26] showed that classification confidence is not well correlated with the localization accuracy, we do not use the confidence score output by the classification head as the basis for whether to update the template. The estimation head produces a feature map $p^{sco}_{x,y} \in \mathbb{R}^{1\times\frac{H_x}{8}\times\frac{W_x}{8}}$.

The classification head is used to evaluate whether the corresponding coordinate position falls within the circle with the radius R of the center of the target. We insist that feature pixels around the object center will have better class estimation quality than edge pixels, so we identify pixels near the object center as foreground samples. The classification head produces a feature map $p^{cls}_{x,y} \in \mathbb{R}^{1\times\frac{H_x}{8}\times\frac{W_x}{8}}$.

Different from the quality estimation head and the classification head, the output dimension of the three-layer perceptron of the bounding localization head is 4, which represent the coordinates (x_0, y_0, x_1, y_1) of the upper left and lower right corners, respectively. Different from anchor-based trackers, which treat the location on the input image as the center of multiple anchor boxes and adjust the anchor points or anchor boxes, the anchor boxes based on prior knowledge may result in unclear matches between anchors and objects. The localization head produces a feature map $B_{x,y} \in \mathbb{R}^{4\times\frac{H_x}{8}\times\frac{W_x}{8}}$.

3.5. Loss Function

We design three loss functions corresponding to three branch tasks: localization quality loss, classification loss, and localization loss. The localization quality score estimation head is adopted to evaluate the IoU score of the bounding box. We only compute the localization quality loss at positive sample locations close to the center of the object, and the loss is defined as:

$$L_{sco} = \frac{1}{N_{pos}} \sum_{x,y} 1_{\{c^*_{x,y}=1\}} * L_{bce}\left(p^{sco}_{x,y}, IoU\left(B_{x,y}, B^*_{x,y}\right)\right) \tag{7}$$

where $1_{\{c^*_{x,y}=1\}}$ is the indicator function that takes 1 if $c^*_{x,y} = 1$ and takes 0 if not. L_{bce} denotes the binary cross-entropy loss, IoU means the IoU score, $B^*_{x,y}$ denote the coordinate $(x^*_0, y^*_0, x^*_1, y^*_1)$ of ground-truth, and N_{pos} is the number of positive samples.

We use the standard focal loss for classification, which is formulated as:

$$L_{cls} = \frac{1}{N_{pos}} \sum_{x,y} L_{focal}\left(p^{cls}_{x,y}, c^*_{x,y}\right) \tag{8}$$

where L_{focal} denotes the focal loss [27] for classification result $p^{cls}_{x,y}$.

Inspired by [28], to obtain more accurate bounding boxes for locations with high classification confidence, we use $p^{cls}_{x,y}$ to dynamically weight the localization loss, which is defined as:

$$L_{loc} = \frac{1}{N_{pos}} \sum_{x,y} 1_{\{c^*_{x,y}=1\}} * L_{IoU}\left(B_{x,y}, B^*_{x,y}\right) * p^{cls}_{x,y} \tag{9}$$

where L_{IOU} denotes the IoU loss [29] for localization result $B_{x,y}$.

We define pixels near the object center as foreground positive samples:

$$c^*_{x,y} = \begin{cases} 1, & if(x,y) - (x^*, y^*) \leq R \\ 0, & otherwise \end{cases} \tag{10}$$

where (x^*, y^*) denote the center coordinates of ground-truth, and R means the radius, which is used to distinguish positive and negative samples.

In general, we optimize the final objective function as follows:

$$L = L_{cls} + \lambda_1 L_{loc} + \lambda_2 L_{sco} \tag{11}$$

where λ_1 and λ_2 are the tradeoff hyperparameters for balancing those three. Both are set to 1 in this work.

3.6. The Mechanism of Updating Dynamic Template

The XYZ response map output from the feature fusion part is sent to the quality score convolution block to calculate the score map, as shown in Figure 7. Select the quality score corresponding to the highest category score position. If the quality score exceeds the threshold, intercept the new dynamic template from the original image to complete the template replacement. Compared with category reliability, the quality score is more suitable as a reference for updating the template. The quality score represents the reliability of the current prediction and reflects the positioning quality of the object in the current image. Better positioning quality ensures that the clipped template has more abundant object information.

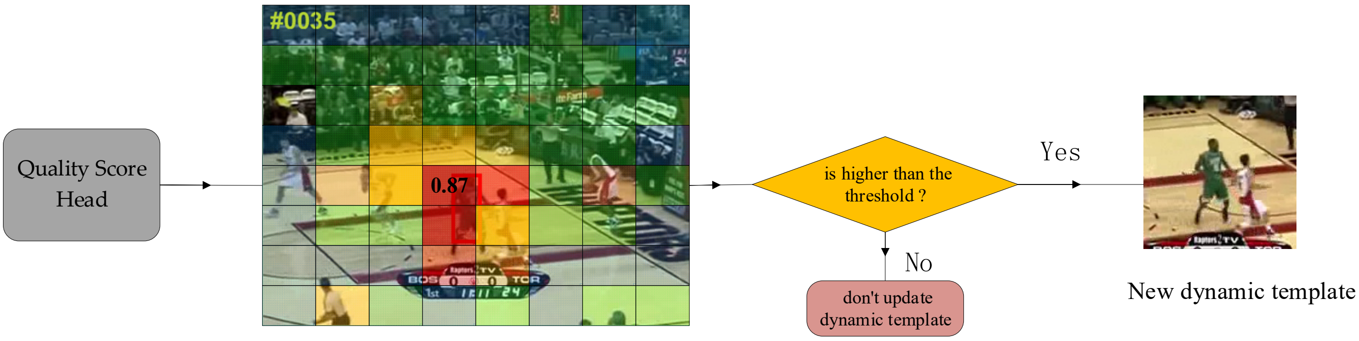

Figure 7. The mechanism of updating the dynamic template.

4. Experiments

This section first introduces the experimental details of our proposed tracker. Then, ablation experiments are presented to analyze the impact of key components in the proposed network. Finally, the results of our tracker on multiple benchmarks are evaluated and our method is compared with other benchmark methods to demonstrate its superiority.

4.1. Implementation Details

This work initializes ResNet-50 [18] with parameters pre-trained on ImageNet [30] and employs it as the backbone network of our tracker. Other parameters of our model are initialized using He initialization [31]. We choose OTB100 [1], GOT10k [32], TrackingNet [33], and LaSOT [34] to compose our base training dataset. Because our network accepts three inputs, for the three video datasets we choose three image frames with an interval of less than 100 as training triples. We employ translation, flipping, and brightness jitter to expand the training set, and translation of tracked objects to edges helps reduce center bias. The sizes of search images, initial templates, and dynamic templates are 256 × 256 pixels, 128 × 128 pixels, and 128 × 128 pixels, respectively. We trained the network for a total of 200 epochs using the AdamW optimizer [35], setting the optimizer's learning rate to 10^{-5}, weight decay to 10^{-4}, and batch size of 64 triples per iteration. The experimental environment is the CPU: i7-9700K GPU: NIDIA TITAN RTX 24G RAM: 32G OS: ubuntu20.04 programming language: python 3.8 main framework: pytorch 1.7, pytracking.

The cross-attention module includes two groups of multi-head attention layers and feedforward network. The number of parallel heads of the multi-header attention layer is set to 8, and the hidden layer dimension of the feedforward network is set to 2048. The feature correlation module between the template and the current search area adopts depth-wise cross correlation [7], which fuses the current frame feature patch with the initial template feature map and the dynamic template feature map, respectively, and then concat the two fusion matrices. The prediction heads consist of a three-layer perceptron of hidden dimension 2048. The dimension of both the quality estimation head and the classification head is 1, except that the output dimension of the localization head is 4. The dynamic template is initialized by the initial template, and in subsequent tracking, it is updated only when the localization quality score of the current prediction result is above the threshold γ, which is set to 0.5. Taking the predicted bounding box as the center, the template-sized image is cropped from the original image as the dynamic template.

4.2. Ablation Study

To verify the efficacy of the proposed components, we perform a component-wise analysis on the GOT-10k benchmark, as presented in Table 1.

Table 1. Ablation study on the GOT-10k test set. Cls means that the classification confidence is directly used as the discriminant condition for updating the dynamic template, Qul means using quality assessment scores as the discriminant condition for updating the dynamic templates, and CA means employing the cross-attention module to enhance the foreground representation of the dynamic template and the search area.

No.	Post Processing	Updating Pipeline	AO
#1	No	No	50.6
#2	Yes	No	56.8
#3	Yes	Cls	58.1
#4	Yes	Qul	58.7
#5	Yes	Qul+CA	61.3

As shown in Table 1, #1 is the baseline model we created, which has the same network architecture as SiamFC++. #1, which has no post-processing, updating template pipeline, or depth-wise correlation structure, has an average overlap (AO) score of 50.6.

When #2 adds post-processing schemes, including cosine window penalty and bounding-box smoothing, the success rate increases significantly by 6.2%. In this way, bounding boxes that are far from the center of the object or whose sizes change drastically will be severely downweighed. Post-processing improves tracking accuracy because the moderated displacement and size changes conform to the motion morphology of objects in the video.

#3 introduces the most simplified updating template pipeline, which directly uses the classification confidence output by the classification head as the discriminant condition for dynamic template update. Get a 128 × 128 size image in the original image from the highest classification score position. The average coincidence rate increased by 1.3% because of the introduction of additional up-to-date object information into the network.

Compared with #3, #4 additionally uses the quality estimation score instead of the classification confidence score as the discriminant condition for a dynamic template update. Since the classification score does not accurately reflect the localization quality of the bounding box, using the quality estimation score can select better prediction results to update the dynamic template. More credible templates with better positioning quality resulted in a 0.6% AO score improvement.

#5 borrows the core idea of the decoder in Transformer and uses the cross multi-head attention module to enhance the foreground object features in the dynamic template and the search area. Benefiting from the enhancement of foreground objects and the filtering of background information, the network can adopt lower update thresholds and even utilize lower quality dynamic templates. Using the initial template to enhance the features of the dynamic template and the search area is also a process of feature pre-fusion. Compared with the traditional correlation operation, this method of feature pre-fusion using attention mechanism can further improve the performance. Benefiting from more credible templates, more frequent updating, and feature pre-fusion between search areas and templates, the AO score of the tracker increased by 2.6%.

4.3. Results and Comparisons

In this subsection, we test the proposed tracker on four benchmarks (OTB-100, GOT-10k, TrackingNet, and LaSOT) and compare it with some state-of-the-art methods.

GOT-10k [32] is a large-scale single-object tracking benchmark consisting of 10k video sequences, covering a variety of common objects. We strictly follow the policy of using only the training subset officially provided by GOT-10k to train the model. We test the tracker on a test subset consisting of 180 video sequences and submit the tracking results to the official online evaluation server. The evaluation indicators, average overlap (AO) and success rate (SR), are published by the evaluation server.

As shown in Table 2 and Figure 8, our tracker (SiamRDT) achieves an AO score of 61.3, an $SR_{0.5}$ score of 72.5, and an $SR_{0.75}$ score of 49.2. SiamRDT outperforms other SiamRPN++ using the same ResNet-50 backbone by 9.5% in the AO score. Compared with SiamFC++ and SiamRPN++, the tracker proposed in this paper introduces cross-attention and dynamic templates, which are more computationally expensive, but still meet the needs of real-time tracking.

Table 2. Comparison of tracking results on GOT-10k benchmark.

	SiamFC [14]	**SiamFCv2 [14]**	**ECO [36]**	**ATOM [11]**	**SiamRPN++ [7]**	**SiamFC++ [6]**	**Ours**
AO	34.8	37.4	31.6	55.6	51.8	59.9	61.3
$SR_{0.5}$	35.3	40.4	30.9	63.4	61.8	69.5	72.5
$SR_{0.75}$	9.8	14.4	11.1	40.2	32.5	47.9	49.2
Fps	44.15	25.81	2.62	20.71	49.83	90	35.26

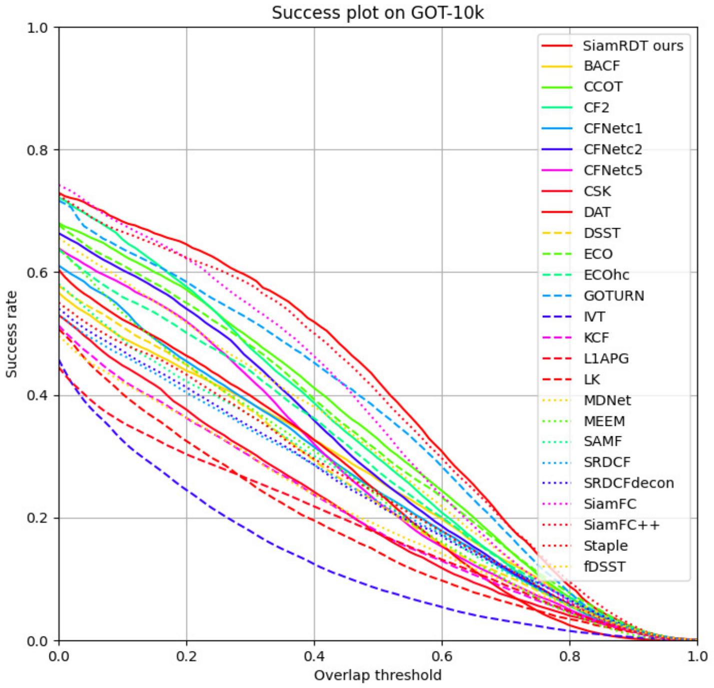

Figure 8. Comparison with other trackers on GOT-10k benchmark.

LaSOT [34] is a large-scale high-quality long-term tracking benchmark consisting of 1400 challenging video sequences. The training set contains 1120 videos and the test set contains 280 videos. It has 70 classes of objects, each class containing 20 videos with an average length of more than 2400 frames.

We follow the one-shot evaluation (OPE) on the LaSOT dataset to compare the success and precision of different trackers. As shown in Table 3, our tracker (SiamRDT) achieves a 56.2 score for success and 56.5 for precision. SiamRDT with a dynamic updating template pipeline outperforms SiamRPN++ by 7.1% and 6.9% in success and precision scores, respectively. The results demonstrate the advantages of trackers with reliable dynamic templates on long-term tracking benchmarks.

Table 3. Comparison of tracking results on LaSOT benchmark.

	SiamFC [14]	**ECO** [36]	**ATOM** [11]	**SiamRPN++** [7]	**SiamBAN** [37]	**SiamFC++** [6]	**Ours**
Succ.	33.9	30.1	50.5	49.1	51.4	54.4	56.2
Prec.	33.6	32.4	51.5	49.6	51.8	54.7	56.5

TrackingNet [33] is a large-scale short-term tracking dataset whose test subset contains 511 sequences covering abundant objects and scenes. It uses an online server to evaluate the tracking results of the tracker on the test split. The precision and success rates of several advanced trackers are shown in Table 4.

Table 4. Comparison of tracking results on TrackingNet benchmark.

	SiamFC [14]	**ECO** [36]	**ATOM** [11]	**D3S** [38]	**SiamRPN++** [7]	**KYS** [39]	**Ours**
Succ.	55.9	55.4	70.3	72.8	73.3	74.0	74.6
Prec.	51.8	49.2	64.8	66.4	69.4	68.8	69.3

As shown in Table 4, it obtains 74.6% and 69.3% in terms of success and precision, respectively. Compared with the online update tracker ATOM, success and precision are improved by 4.3% and 4.5%, respectively. It shows that the proposed method also exhibits certain advantages on short-term datasets.

The **OTB100** contains 100 challenging video sequences. It focuses on testing and analyzing the ability of the tracker to deal with different scenes, such as illumination variation, deformation, occlusion, and fast motion. The evaluation is mainly based on two criteria: precision and success rates. The accuracy measures the distance between the tracking result and the ground-truth center, while the success rate measures the overlap between the estimation boxes and the ground-truth boxes. The precision and success rates of several advanced trackers are shown in Table 5.

Table 5. Comparison of tracking results on OTB100 benchmark.

	SiamFC [14]	**CFNet** [40]	**SRDCF** [41]	**SiamDWfc** [42]	**SiamRPN** [15]	**Ours**
Succ.	72.3	72.4	72.5	78.2	81.5	84.5
Prec.	76.5	77.4	78.7	82.3	84.9	90.5

As shown in Table 5 and Figure 9, it obtains 84.5% and 90.5% in terms of success and precision, respectively. Compared with the SiamRPN, success and precision are improved by 3% and 5.6%, respectively. As shown in Figure 10, our model shows excellent performance in dealing with illumination variation, deformation, occlusion, fast motion, and complex background.

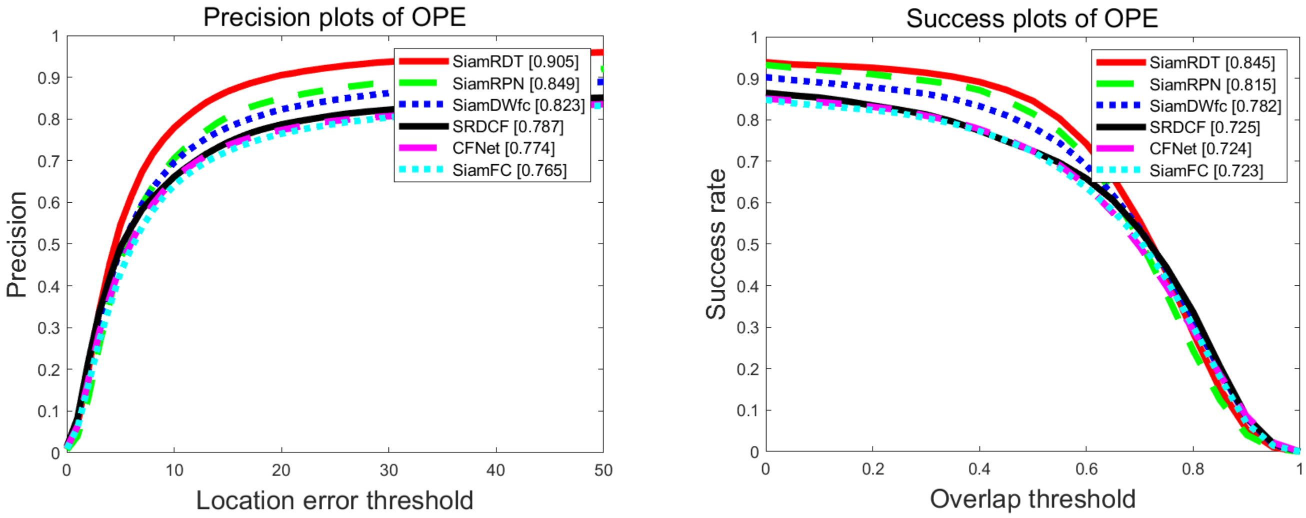

Figure 9. Comparison with other trackers on OTB100 benchmark.

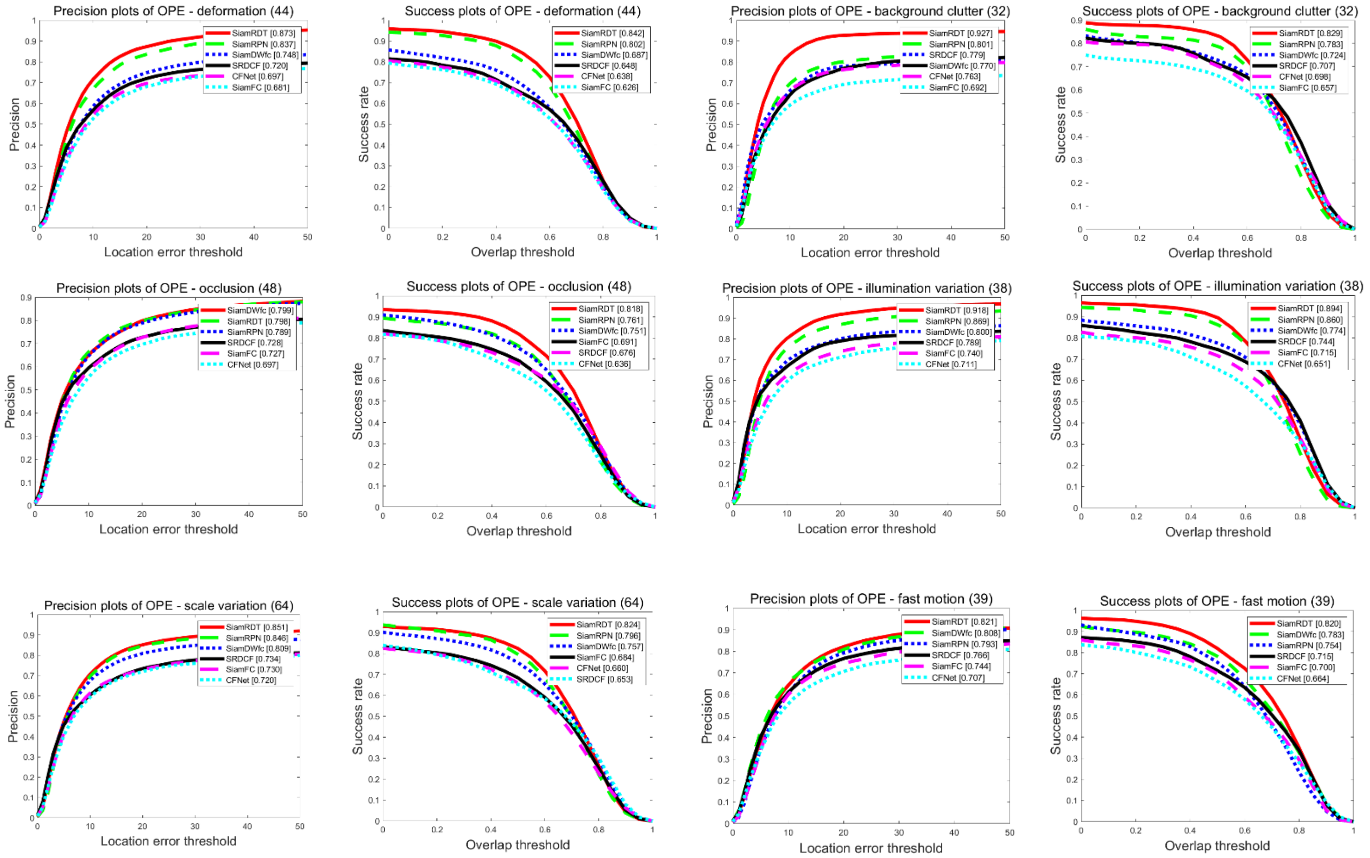

Figure 10. Comparison with other trackers on the OTB100 benchmark in the case of deformation or background clutter or occlusion or illumination variation or scale variation or fast motion, respectively. The comparison results show that the latest state information of the object provided by the additional dynamic template can help the tracker adapt to complex environmental changes and increase the robustness of the tracker.

4.4. *Visualization of Tracking Results*

In this subsection, we test and compare the proposed method with several advanced methods on several challenging video sequences. The tracking results of the model under interference factors such as illumination variation, deformation, occlusion, fast motion, and complex background are shown in Figure 11.

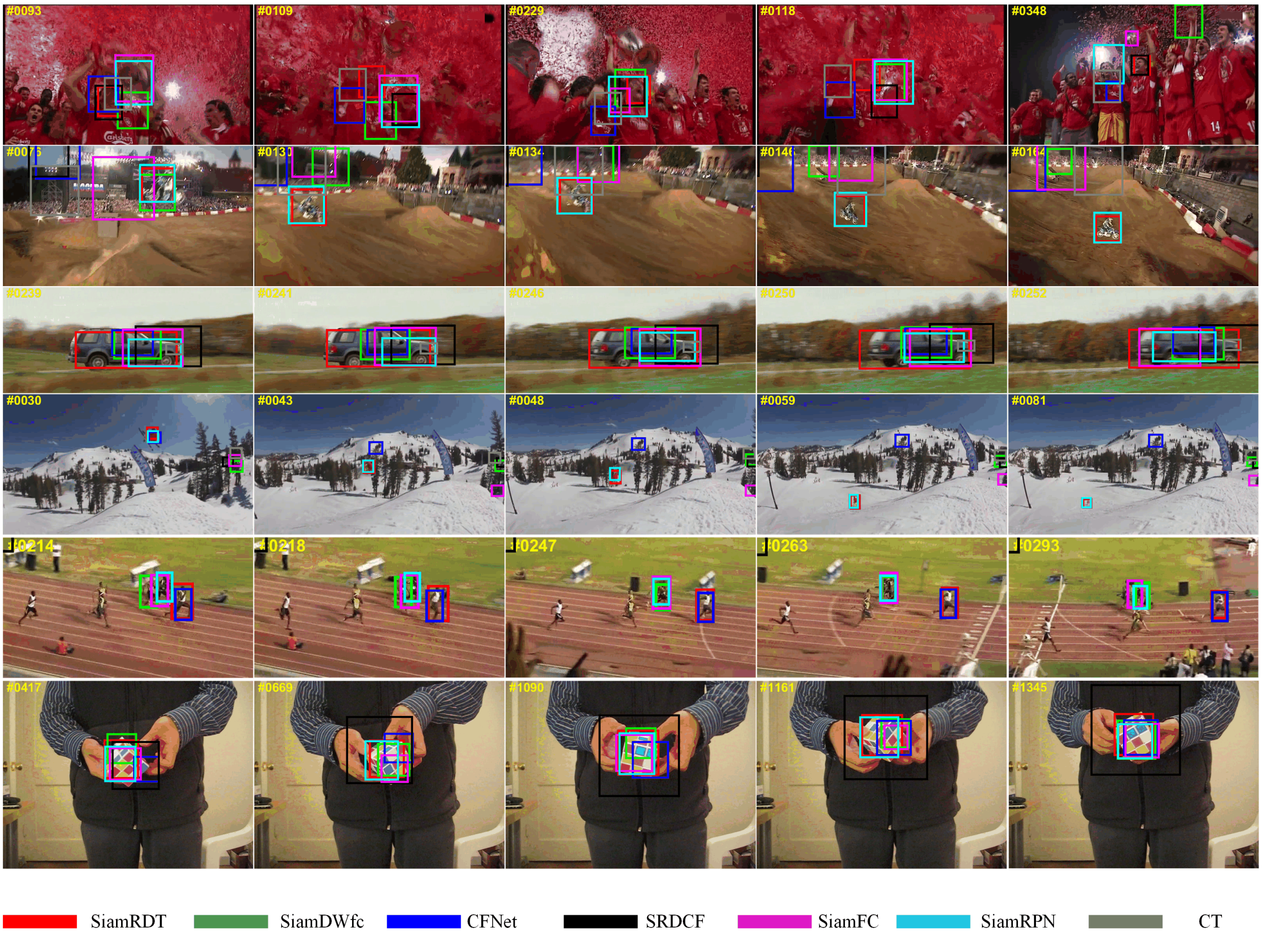

Figure 11. The visual results show that the proposed tracker can track stably under a variety of complex conditions.

As shown in Figure 12, including our proposed tracker, we lose targets when they respond to rapid, irregular motion of the target. This is mostly due to post-processing operations, where the model tends to generate higher weights near the location of the last prediction. In addition, to reduce the computational cost and improve the prediction speed, the size of the search area of most trackers including ours does not cover the entire image. Most of the objects in the training dataset appear in the center of the image, and we apply random cropping during training to try to avoid the positional bias of the tracker.

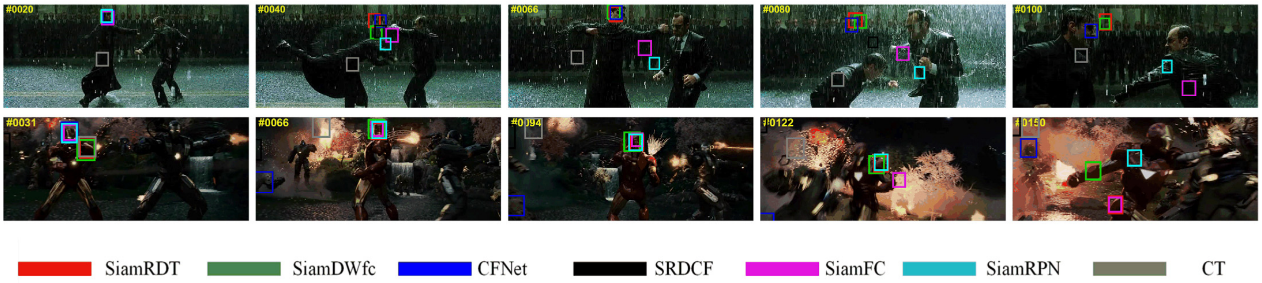

Figure 12. The visual results show that some trackers, including our proposed tracker, lose targets when they respond to rapid, irregular motions of the target.

5. Conclusions and Future Work

In this paper, we construct a feature enhancement pipeline based on the attention mechanism and propose a novel anchor-free object tracker with symmetric structure based on a reliable dynamic template. The feature representation of the dynamic template and the search area is enhanced by using a cross-template attention mechanism. Through reliable dynamic templates and credible initial templates, the model can fuse initial-state information and the latest-state information of objects, which enables our model to combine temporal and spatial information to improve its performance. Elaborate ablation studies have demonstrated the effectiveness of key components of the model. Extensive experimental results on many benchmarks (OTB100, GOT10k, LaSOT, and TrackingNet) show that the proposed tracker significantly outperforms some state-of-the-art algorithms and operates at real-time speeds. In future work, the single dynamic template will be expanded into a multi-template queue to completely record the state changes of objects over time for further improve the long-term tracking effect. In addition, the multi-template queue will record templates of various objects, extending single-object tracking to multi-object tracking.

Author Contributions: Conceptualization, Q.Z. and Z.W.; methodology, Z.W.; software, Z.W.; validation, Z.W., Q.Z. and H.L.; formal analysis, Q.Z.; investigation, Q.Z.; resources, H.L.; data curation, Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, Z.W.; visualization, Z.W.; supervision, Q.Z.; project administration, Q.Z.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Science Foundation of Shandong Province grant number ZR2020MF005.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data was obtained from http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html. http://got-10k.aitestunion.com/. https://tracking-net.org/.

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

References

1. Wu, Y.; Lim, J.; Yang, M.-H. Online Object Tracking: A Benchmark. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
2. Zhang, K.; Zhang, L.; Yang, M.-H. Fast Compressive Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2002–2015. [CrossRef] [PubMed]
3. Galoogahi, H.K.; Fagg, A.; Lucey, S. Learning Background-Aware Correlation Filters for Visual Tracking. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1144–1152.
4. Li, X.; Hu, W.; Shen, C.; Zhang, Z.; Dick, A.; Hengel, A.V.D. A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol. (TIST)* **2013**, *4*, 1–48. [CrossRef]
5. Smeulders, A.W.M.; Chu, D.M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual Tracking: An Experimental Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1442–1468. [PubMed]
6. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12549–12556.
7. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
8. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-Aware Anchor-Free Tracking. In Proceedings of the European Conference on Computer Vision—ECCV 2020: Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 771–787.
9. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 8126–8135.
10. Bhat, G.; Danelljan, M.; Van Gool, L.; Timofte, R. Learning Discriminative Model Prediction for Tracking. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6182–6191.
11. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 4655–4664.

12. Yang, T.; Chan, A.B. Learning Dynamic Memory Networks for Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 153–169.
13. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese Instance Search for Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429.
14. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; Volume 9914, pp. 850–865.
15. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
16. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H.S. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1328–1338.
17. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *NIPS* **2012**, *60*, 84–90. [CrossRef]
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
19. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European conference on computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
20. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6568–6577.
21. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully convolutional one-stage object detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9626–9635.
22. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
23. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Volume 12346, pp. 213–229.
24. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
26. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of Localization Confidence for Accurate Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 784–799.
27. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
28. Peng, J.; Jiang, Z.; Gu, Y.; Wu, Y.; Wang, Y.; Tai, Y.; Wang, C.; Lin, W. SiamRCR: Reciprocal Classification and Regression for Visual Object Tracking. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, Valletta, Malta, 25–27 October 2021; Volume 1, pp. 952–958.
29. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
30. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the International Conference on Computer Vision, Las Condes, Chile, 11–18 December 2015; pp. 1026–1034.
32. Huang, L.; Zhao, X.; Huang, K. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1562–1577. [CrossRef] [PubMed]
33. Müller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 310–327.
34. Fan, H.; Ling, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5369–5378.
35. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
36. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939.
37. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese Box Adaptive Network for Visual Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6667–6676.
38. Lukezic, A.; Matas, J.; Kristan, M. D3S—A Discriminative Single Shot Segmentation Tracker. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7131–7140.

39. Bhat, G.; Danelljan, M.; Van Gool, L.; Timofte, R. Know your surroundings: Exploiting scene information for object tracking. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 205–221.
40. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.S. End-to-End Representation Learning for Correlation Filter Based Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern, Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008.
41. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
42. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4586–4595.