TGLC: Visual object tracking by fusion of global-local information and channel information

Shuo Zhang[1], Dan Zhang*[2], Qi Zou[1]
[1] Lassonde School of Engineering, York University, Toronto, ON M3J 1P3, Canada
[2] Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

*Corresponding author:
Dan Zhang, Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
E-mail address: dan.zhang@polyu.edu.hk

TGLC: Visual object tracking by fusion of global-local information and channel information

Abstract

Visual object tracking aspires to locate the target incessantly in each frame with designated initial target location, which is an imperative yet demanding task in computer vision. Recent approaches strive to fuse global information of template and search region for object tracking, which achieve promising tracking performance. However, fusion of global information devastates some local details. Local information is essential for distinguishing the target from background regions. With a focus on addressing this problem, this work presents a novel tracking algorithm TGLC integrating a channel-aware convolution block and Transformer attention for global and local representation aggregation, and for channel information modeling. This method is capable of accurately estimating the bounding box of the target. Extensive experiments are conducted on five widely recognized datasets, i.e., GOT-10k, TrackingNet, LaSOT, OTB100 and UAV123. The results depict that the proposed tracking method achieves competitive tracking performance compared with state-of-the-art trackers while still running in real-time. Visualization of the tracking results on LaSOT further demonstrates the capability of the proposed tracking method to cope with tracking challenges, e.g., illumination variation, deformation of the target and background clutter.

Keywords Visual object tracking · Global-local representation aggregation · Channel information · Transformer attention · Convolution

1. Introduction

Visual object tracking is one of the most important and cutting-edge topics in robot vision and computer vision, which aims at locating the target continuously with given object position in the initial frame. Visual object tracking has been widely utilized in many applications such as autonomous vehicles [1], human machine interaction [2], video surveillance [3] and robot perception [4], *etc*. There are a number of tracking challenges such as illumination variation, appearance variation, fast motion, motion blur, out-of-plane rotation, scale variation and occlusion, *etc*. [5], which cause object tracking an extremely difficult task.

Object tracking methods are generally categorized into generative methods and discriminative methods [6]. Generative methods [7, 8, 9, 10] mainly analyze and model the object area in the current frame and find the area (the next object location) that best resembles the model in the next frame. Discriminative methods [11, 12] are increasingly popular, which usually train classifiers to distinguish the object location from background regions. Siamese-based trackers gradually become the dominant discriminative approaches. Representative Siamese-based trackers usually consist of a shared backbone for feature extraction, a feature fusion module and one or more prediction head(s).

Numerous feature fusion methods are employed for Siamese-based trackers, which are divided into two main categories, cross correlation-based methods and attention-based fusion approaches. Naive correlation is a simple method for correlation calculation, which is commonly used in previous Siamese trackers such as [13, 14, 15]. Depth-wise correlation calculates correlation of

template and search image in a sliding window at each depth level [16]. Pixel-wise correlation [17, 18] computes feature connection in smaller areas than naive or depth-wise correlation, which better maintains the spatial features of the search region. Correlation-based methods solely establish locally linear similarity relations between search region and template, which are less discriminative and less informative. Therefore, trackers of this type are greatly limited in performance. Recently, some researchers [19, 20] start to exploit attention-based networks to fuse features. These approaches focus more on specific features, achieving better tracking results than correlation-based fusion methods. However, attention-based feature fusion methods [21] strive to build global associations of input feature maps without modeling connections between local representations. Locally detailed features are especially important when tracking challenges occur. For example, visible partial information can be utilized to locate the target when the target is partially occluded. Therefore, it is quite essential to simultaneously model the global and local representations for more accurate feature interaction. In addition, most existing feature fusion methods only exploit spatial information of feature maps, which fail to exploit channel representations. Lack of channel information results in low object localization ability.

In this paper, a novel tracking algorithm (TGLC) is developed to tackle the forenamed problems. This tracking method is composed of a shared backbone for feature extraction, a novel feature fusion module and a key point prediction head. The proposed feature fusion module consists of a number of global-local self-attention (GLSA) and global-local cross-attention (GLCA) modules. Each GLSA or GLCA contains a self-attention or cross-attention module for global information capturing, a channel-aware convolution block (CCB) for modeling local feature information and channel information. This feature fusion method can improve the target localization ability and bounding box prediction capability of the proposed tracking algorithm.

The main contributions of the proposed method are shown below.

- A channel-aware convolution block (CCB) is applied behind each multi-head self-attention and cross-attention in the feature fusion module. This design fully captures global and local information of two feature maps.
- Another superiority of the proposed feature fusion module is its ability to perceive channel information of feature maps. Both spatial attention and channel attention are applied to feature maps for modeling essential spatial information and channel information.
- The proposed tracker (TGLC) achieves comparable and even better tracking performance compared with state-of-the-art tracking methods on several widely utilized tracking datasets, e.g., GOT-10k [22], TrackingNet [23], LaSOT [24], OTB100 [25] and UAV123 [26]. The tracking speed of the proposed tracker is approximately 50 fps on a single RTX A6000 GPU, which is suitable for real-time applications.

2. Related Work

Visual object tracking. SiamFC [13] is one of the most conventional Siamese-based tracking methods. It adopts a fully convolutional network to predict tracking results. This method improves the ability for tracking unpredictable targets. SiamRPN [14] incorporates the region proposal network with the Siamese tracking framework. Cross correlation is utilized to fuse features of two input branches. DaSiamRPN [15] proposes an effective strategy to identify distractors. It learns more discriminative features, thereby significantly improving the tracking performance on extensive long-term and short-term tracking benchmarks. SiamMask [16] innovatively integrates object tracking and segmentation by exploiting a fully convolutional Siamese network. This method replaces the traditional naive correlation with a depth-wise correlation and achieves state-

of-the-art performance both on object tracking and segmentation datasets. Alpha Refine [18] designs a highly transferable refinement module for more precise bounding box estimation and adjustment. It employs pixel-wise correlation and a corner estimation head to fully retain spatial features. In order to exhaustively interact the representations of the template and search region, Transformer-based feature fusion schemes are proposed in recent years. TransT [19] designs a purely Transformer attention-based feature integration module, which outperforms the existing correlation-based trackers substantially. STARK [20] concatenates the search and template representations and feeds them into a Transformer encoder-decoder module for interaction. A dynamic template is utilized to provide timely updated representations of the target.

Attention-related feature fusion modules have global receptive fields, which focus on holistic information of the two input feature maps. Despite their superior performance, the lack of modeling local details hinders the further improvement of these methods. Motivated by the feature fusion structure of TransT [19], a novel feature interaction module is developed in this work to fully model global and local representations of input features. The main differences of the feature fusion modules between the proposed TGLC and TransT are two-fold. (1) TransT designs a pure attention module for feature fusion. TGLC combines the Transformer attention and convolutions for global and local feature aggregation. (2) TransT solely exploits spatial information of the feature maps. However, TGLC incorporates both spatial and channel information for more comprehensive feature fusion. The proposed feature fusion module improves the tracking results.

Integration of global and local information. BoTNet [27] proposes a Bottleneck Transformer (BoT) block to build a new backbone framework, which incorporates self-attention into the bottleneck block of ResNet. This method is capable of extracting global and local feature information and achieves state-of-the-art performance in object detection, instance segmentation and image classification. Xu *et al*. [28] present a co-scale image classification method, which utilizes depth-wise convolutions to build position encodings and relative position encodings for transformer attentions. A great number of serial and parallel modules are employed to estimate multiple scales. This proposed method outperforms existing attention-based or convolution-based classification approaches. CoAtNet [29] introduces an efficient combination of ConvNets and Transformer attentions, which retains superior generalization ability and model aptness. This algorithm realizes state-of-the-art classification performance. Conformer [30] develops a double-branched structure with convolution network and attention network in separate branches. Global feature representations and locally detailed features are communicated in a feature coupling block. This method leverages the advantages of both convolution and Transformer attention, retaining exceptional performance for classification and detection. Mehta *et al*. [31] design a light-weight network combining CNNs and transformer for light mobile devices. This network learns global representations with time-efficient transformer and demonstrates admirable performance on ImageNet and object detection datasets.

Woo *et al*. [32] develop a network integrating channel attention and spatial attention mechanism, which is suitable for almost all ConvNets. It considerably improves the detection and classification ability. An advantageous algorithm is introduced in [33] for object detection of UAV images by integrating CNNs and transformer attention. The integrated algorithm achieves advanced detection performance. Zhang *et al*. [34] present an object detection method by adopting deep convolution network as backbone and attention mechanism for emphasizing important features. A cell segmentation approach is developed in [35] by fusing features of CNNs with representations from Transformer attention, which improves the segmentation mIoU score. U-Net Transformer [36] is proposed for precise medical image segmentation, which employs self-

attention and cross-attention to enhance the conventional U-Net for establishing long-range relations. This method brings improvements to segmentation accuracy.

The aforementioned methods are capable of leveraging the advantages of attention and convolutions. Global and local feature representations are established for improving the performance of vision tasks. However, most approaches are developed for building a general backbone architecture for vision tasks, e.g., image classification, instance segmentation and object detection. Our proposed tracking algorithm (TGLC) is differentiated from above methods by developing a novel feature fusion module incorporating global long-range representations and locally detailed features of images. Channel information is also exploited in the feature fusion module to improve the tracking performance.

3. Method

3.1. Overall Tracking Framework

The proposed overall tracking framework is shown in Figure 1, which consists of a shared backbone for feature extraction of template and search region, a feature fusion module to aggregate backbone features of two branches, and a key point prediction head for generating bounding boxes. The first four convolution blocks of ResNet50 [37] are exploited to build the backbone. In order to retain more detailed feature information, the stride of the down-sampling of the fourth block is set as 1. Additionally, a dilated convolution with a dilation rate of 2 is employed to replace the second convolution of the fourth block to obtain a wider field of view without increasing the computational cost. The accumulated stride of the backbone is 8. The template and search region are separately cropped from the first frame and current frame. Their backbone features are further extracted by the shared backbone.

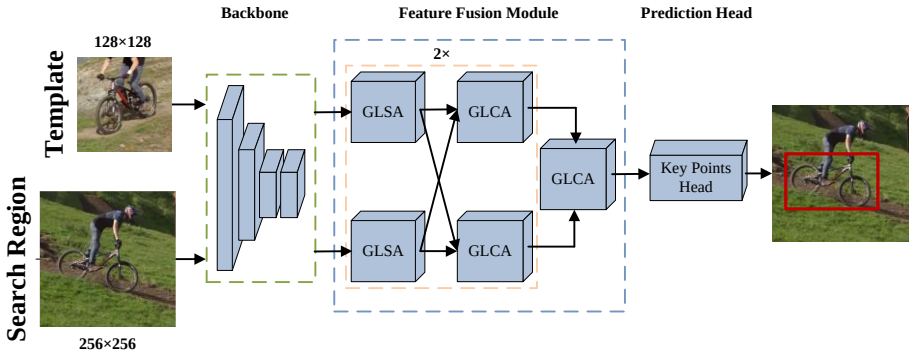


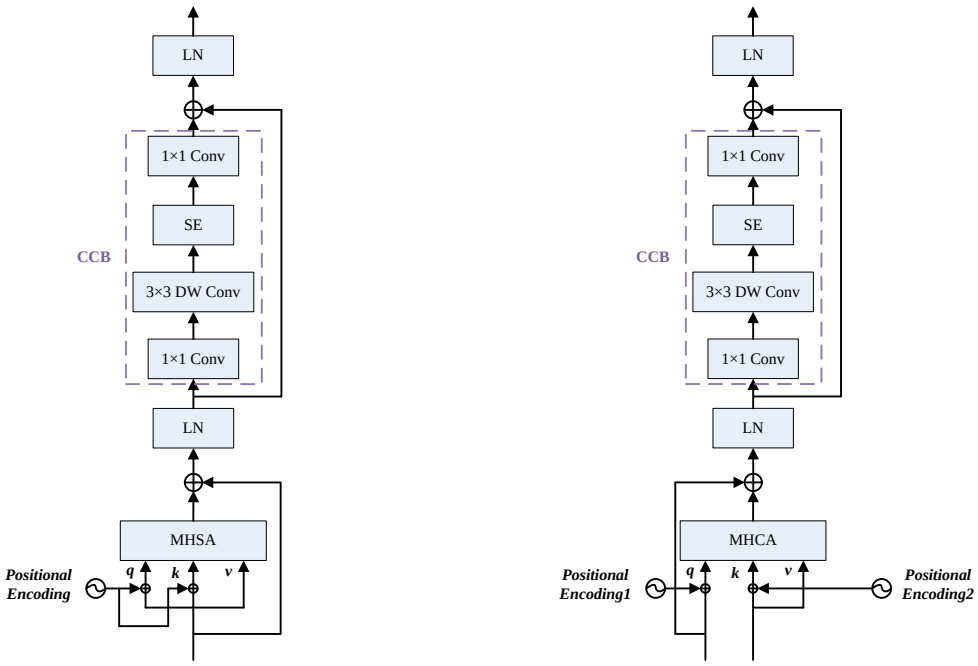
Figure 1 Structure of the proposed tracking method

3.2. Global Local Feature Fusion

$I_{t1} \in \mathbb{R}^{[C_1, \frac{H_t}{8}, \frac{W_t}{8}]}$ and $I_{s1} \in \mathbb{R}^{[C_1, \frac{H_s}{8}, \frac{W_s}{8}]}$ respectively represent the backbone features of template and search region, where C_1 is the channel number (1024). H_t, W_t, H_s and W_s denote separately heights and widths of the template and search region. These two backbone features are dimensionally reduced to 256 channels for the purpose of improving computational efficiency. Then they are flattened spatially, resulting in two new features, $I_{t2} \in \mathbb{R}^{[C_2, \frac{H_t}{8} \times \frac{W_t}{8}]}$ and $I_{s2} \in \mathbb{R}^{[C_2, \frac{H_s}{8} \times \frac{W_s}{8}]}$. C_2 is the new channel number (256). Flattening is necessary for Transformer attention since flattened feature maps are more computationally efficient than original feature maps for attentions.

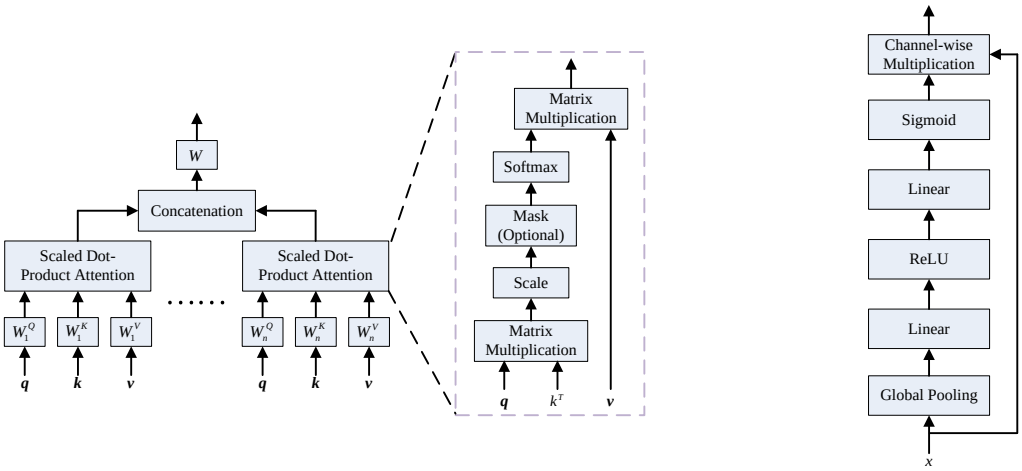
However, flattening operation will cause the loss of one dimension of images, which will lose the spatial connections among pixels. In order to solve this problem, positional encoding is incorporated and added to the original feature maps before flattening. Positional encoding generated by sine and cosine functions [38] is utilized in our feature fusion module. After dimension reduction and feature flattening, I_{t2} and I_{s2} are fed into the feature fusion module.

The feature fusion module is demonstrated in Figures 1, 2 and 3. Self-attention and cross-attention merely focus on global information of feature maps, ignoring the local detailed information. Local features are especially important when there are tracking challenges such as similar distractors. In order to tackle this issue, a channel-aware convolution block (CCB) is applied behind each multi-head self-attention and cross-attention. The block consists of a 1×1 convolution, a 3×3 depth-wise convolution, a squeeze-and-excitation layer (SE layer) [39] and a 1×1 convolution. Convolutions are capable of capturing local features of targets with better translation invariance and generalization ability, which can be exploited to complement the deficiencies of transformer attentions.



(a) GLSA (b) GLCA

Figure 2 Structures of GLSA and GLCA



(a) MHSA and MHCA (b) SE in CCB

Figure 3 Structures of MHA and SE in CCB

Figures 2 and 3 show the detailed structures combining attention and convolutions in global-local self-attention (GLSA) and global-local cross-attention module (GLCA). The difference between GLSA and GLCA is that the query (q), key (k) and value (v) in GLSA are from a single branch, while they are from two branches in GLCA. The first block in GLSA and GLCA is multi-head attention, which includes multi-head self-attention (MHSA) and multi-head cross-attention (MHCA). The structures of MHSA and MHCA are identical, while the query (q), key (k) and value (v) follow the difference of GLSA and GLCA. Multi-head attention comes from scaled dot-product attention, the scaled dot-product attention is described in Eq. (1).

$$Att(q,k,v) = \text{softmax}(qk^T(d_k)^{(-\frac{1}{2})})v \tag{1}$$

where q,k,v are three matrices and $q \in \mathbb{R}^{n_q \times d_k}$, $k \in \mathbb{R}^{n \times d_k}$, $v \in \mathbb{R}^{n \times d_v}$. n_q and n are respectively the sequence length for the query and the key. d_k and d_v denote dimensions for the key and the value, $d_k = d_v$ in this case. T denotes transpose of a matrix. Softmax function is used to map all elements into $(0,1)$. For i^{th} element $f_i\big|_{i\in\Omega}$, the softmax value of f_i is shown in Eq. (2).

$$\text{softmax}(f_i) = \frac{\exp(f_i)}{\sum_{j\in\Omega}\exp(f_j)} \tag{2}$$

Multi-head attention exploits multiple scaled dot-product attentions to obtain information from numerous subspaces [21]. The calculation process is demonstrated in Figure 3(a). q,k,v are linearly transformed into different inputs for several single-head attentions by n sets of linear transformation matrices W_i^Q, W_i^K and W_i^V, $i \in n$, where n denotes the number of heads. Outputs from different heads are concatenated and then transformed by another linear transformation matrix W. The process is calculated in Eq. (3) and Eq. (4).

$$MH\_Att(q,k,v) = \text{Concat}(Head_1,\ldots,Head_n)W \tag{3}$$

$$Head_i = Att(qW_i^Q, kW_i^K, vW_i^V) \tag{4}$$

The input of multi-head attention is added to the output to form a residual connection, which

can alleviate gradient vanishing and network overfitting problems. There are two cases in MHSA of GLSA, which results from two input branches, the template features I_t and search region features I_s. MHSAs of search region branch and template branch in a residual form are illustrated in Eq. (5) and Eq. (6), respectively.

$$Y_s = I_s + MH\_Att(q = I_s + \Gamma_s, k = I_s + \Gamma_s, v = I_s) \tag{5}$$

$$Y_t = I_t + MH\_Att(q = I_t + \Gamma_t, k = I_t + \Gamma_t, v = I_t) \tag{6}$$

where Γ_s and Γ_t are position encodings for I_s and I_t. There are also two types of MHCAs in GLCA where the query is from one branch, the key and the value are from the other branch, which are demonstrated in Eq. (7) and Eq. (8).

$$Y_{s\_t} = I_s + MH\_Att(q = I_s + \Gamma_s, k = I_t + \Gamma_t, v = I_t) \tag{7}$$

$$Y_{t\_s} = I_t + MH\_Att(q = I_t + \Gamma_t, k = I_s + \Gamma_s, v = I_s) \tag{8}$$

Layer Normalization (LN) is utilized after multi-head attention for speeding up training and convergence of the model. Eq. (9) demonstrates the calculation process for LN, where x is the input with the mean μ and standard deviation δ. N and ε denote the number of channels of x and a small positive value to prevent a denominator of 0, respectively. α and β are learnable scaling parameters.

$$LN(x) = \alpha \cdot \frac{x - \mu}{\delta} + \beta \tag{9}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{10}$$

$$\delta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 + \varepsilon} \tag{11}$$

The next module is the channel-aware convolution block (CCB), which is employed to capture local feature information, complementing with the multi-head attentions that focus on global feature information. There is another superiority of this module, which is the ability to perceive channel information of the feature maps. This superiority is achieved by SE block shown in Figure 3(b). Convolutions and attentions can only exploit spatial information of the feature maps, which fail to leverage feature channel information. The use of SE block in CCB provides plentiful information of different channels for tracking models.

Assuming that x is the input of CCB, which is reshaped into a 3D feature map $\tilde{x}$ with shape of $[\tilde{C}, \tilde{H}, \tilde{W}]$. $\tilde{C}$, $\tilde{H}$ and $\tilde{W}$ denote channel number, height, and width of $\tilde{x}$, respectively. $\tilde{C} = 256$, $\tilde{H} = \tilde{W} = 16$ for template branch, $\tilde{H} = \tilde{W} = 32$ for search region branch. The first part of CCB is a convolution with kernel size of 1, which is utilized to expand the channel dimension from 256 to 1024. The convolution is followed by a batch normalization and a ReLU nonlinear activation function. This 1×1 convolution is depicted in Eq. (12).

$$h[n,k,l] = \sum_{m}\sum_{i=1}\sum_{j=1} K[m,n,i,j] \cdot \tilde{x}[m, k+i-1, l+j-1] \tag{12}$$

where K is the convolution kernel, h is the convolution output. $m = 256$, $n = 1024$ denote the

number of input channels and output channels, respectively. $[k,l]$ and $\cdot$ represent the coordinates of any point in $\tilde{x}$ and multiplication. Batch normalization is similar to layer normalization, the difference is that N denotes batch size for batch normalization in Eq. (10) and Eq. (11). ReLU is defined as Eq. (13).

$$ReLU(x) = \max(0, x) \tag{13}$$

The second part of CCB is a 3×3 depth-wise convolution, which is demonstrated in Eq. (14). $\widehat{K}$ denotes the convolution kernel with size of 3. The channels of the kernel $\widehat{K}$, the feature map $\tilde{x}$ and the output map h have a one-to-one correspondence in depth-wise convolutions, which makes them simpler and more efficient than regular convolutions. A batch normalization and a ReLU activation function are used after the depth-wise convolution.

$$h[n,k,l] = \sum_{n} \sum_{i=0}^{2} \sum_{j=0}^{2} \widehat{K}[n,i,j] \cdot \tilde{x}[n,k+i-1,l+j-1] \tag{14}$$

The third part of CCB is the SE block, as shown in Figure 3(b). Input x is first fed into a global average pooling module, as demonstrated by Eqs. (15-17). This module calculates the average value y_t along two spatial dimensions W and H of x_t, where x_t is the feature map of t^{th} channel of x. y and C denote the output and the number of channels in x, respectively. Two linear layers (FC layers), W_1 and W_2, are employed after global pooling to build weights of channels, as shown in Eq. (18). $\Re$ is a ReLU activation function. $\text{Sigmoid}\left[W_2\left(\Re(W_1 y)\right)\right]$ is the final channel representation, which is applied to the input x as channel attention weights by channel-wise multiplication. $\hat{y}$ is the output of SE block.

$$x = \left\{ x_t \Big|_{t \in \{1,2,\ldots,C\}} \right\} \tag{15}$$

$$y_t = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} x_t(i,j) \tag{16}$$

$$y = \left\{ y_t \Big|_{t \in \{1,2,\ldots,C\}} \right\} \tag{17}$$

$$\hat{y} = \text{Sigmoid}\left[W_2\left(\Re(W_1 y)\right)\right] \cdot x \tag{18}$$

$$\text{Sigmoid}(t) = \frac{1}{1+\exp(-t)} \tag{19}$$

The final part of CCB is another 1×1 convolution, followed by a batch normalization layer. The channel number decreases from 1024 to 256. The input of CCB is added to the output to form a residual connection. Then, a layer normalization is utilized to the residual output of CCB, which generates the final output of the GLSA and GLCA.

3.3. Key Point Prediction Head

The fused features are fed into the key point prediction head to generate the bounding boxes for the object. The prediction head is inspired by [40], which mainly predicts the required key points for construction of the bounding boxes. The main differences with [40] are two-folds. (1) Depth-wise correlation adopted in [40] is not utilized in our prediction head. (2) Only features from the last scale are exploited in our prediction head, while multiscale features are utilized in the prediction head of [40]. The detailed structure of the key point prediction head is demonstrated

in Figure 4, which consists of a regression branch and a classification branch. Two 3×3 convolution blocks are employed for two branches, respectively. Each convolution block is composed of a 3×3 convolution shown in Eq. (20), a batch normalization and a ReLU activation function. Eq. (20) resembles Eq. (14), while the difference is that channels of kernel $\bar{K}$ and input ζ are not necessarily corresponding. p and q denote respectively the number of input channels and that of output channels, which are both 256 and remain unchanged for all the components of the prediction head. For the regression branch, another 3×3 convolution block and a 1×1 convolution are used to predict coordinates of the first set of key points $kp1$. As shown in Eq. (21), the offset $(\Delta i, \Delta j)$ is calculated by $kp1$ and $base\_offset$, ($base\_offset$ is an initial set of points with values between -1 and 1).

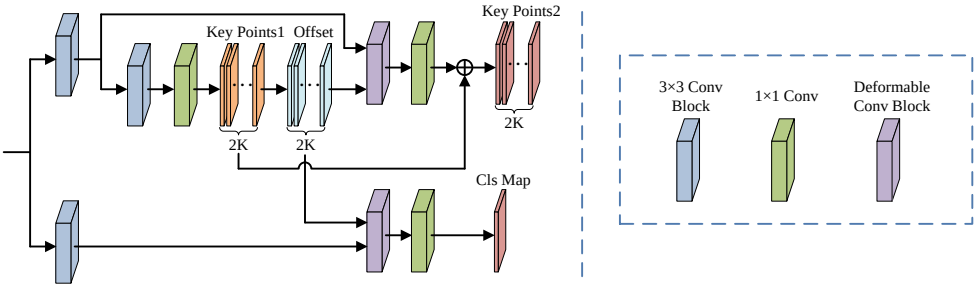


Figure 4 Structure of key point prediction head

$$h[q,k,l]=\sum_{p}\sum_{i=0}^{2}\sum_{j=0}^{2}\bar{K}[p,q,i,j]\cdot\zeta[p,k+i-1,l+j-1] \tag{20}$$

$$(\Delta i,\Delta j)\big|_{offset}=kp1-base\_offset \tag{21}$$

Then the estimated offset $(\Delta i, \Delta j)$ and feature map $\hat{\zeta}$ are fed into a deformable convolution block, which includes a 3×3 deformable convolution shown in Eq. (22), a batch normalization and a ReLU activation. The offset is added to the regular sampling locations to expand the receptive fields of original convolutions, which is a superiority of deformable convolutions. Nevertheless, positions after the offset might not be integers, which is tackled by bilinear interpolation. The interpolation process is manifested by Eq. (23), where $s_x=k+i-1+\Delta i$, $s_y=l+j-1+\Delta j$, denote the coordinates of a random non-integer position. $[t_x^z,t_y^z]$ denotes the coordinates of one of integer positions in $\hat{\zeta}$, where N represents a collection of all integer positions of $\hat{\zeta}$. Eq. (24) is utilized to calculate the g in Eq. (23).

$$\hat{h}[q,k,l]=\sum_{p}\sum_{i=0}^{2}\sum_{j=0}^{2}\hat{K}[p,q,i,j]\cdot\hat{\zeta}[p,k+i-1+\Delta i,l+j-1+\Delta j] \tag{22}$$

$$\hat{\zeta}[p,s_x,s_y]=\sum_{z\in N}g(s_x,t_x^z)\cdot g(s_y,t_y^z)\cdot\hat{\zeta}[p,t_x^z,t_y^z] \tag{23}$$

$$g(u,v)=\max\left(1-|u-v|,0\right) \tag{24}$$

Following the deformable convolution block, a 1×1 convolution is further exploited to map the number of output channels to $2K$. ($K=9$, denotes the number of key points). The output of the 1×1 convolution is added to the first set of key points $kp1$ to obtain the second group of key points $kp2$. The classification branch also utilizes a deformable convolution block and a 1×1 convolution

to estimate the final classification map (Cls Map), which is for foreground-background classification.

In order to be consistent with the training labels, the moment-based method [41] is utilized to transform the key points $kp1$ and $kp2$ into bounding boxes. A is assumed to be a set of predicted key points. The coordinates of the top-left corner (x_{tl}, y_{tl}) and bottom-right corner (x_{br}, y_{br}) of the bounding box are calculated by Eqs. (26) and (27), respectively. μ_x, δ_x, μ_y and δ_y are respectively the mean values and standard deviations for all x_n and y_n in A. β_x and β_y are two learnable parameters, which are used to adaptively adjust the scales of the transformed bounding boxes.

$$A = \left\{ (x_n, y_n) \Big|_{n \in \{1,2,\ldots,K\}} \right\} \tag{25}$$

$$(x_{tl}, y_{tl}) = (\mu_x - \delta_x \cdot e^{\beta_x}, \mu_y - \delta_y \cdot e^{\beta_y}) \tag{26}$$

$$(x_{br}, y_{br}) = (\mu_x + \delta_x \cdot e^{\beta_x}, \mu_y + \delta_y \cdot e^{\beta_y}) \tag{27}$$

3.4. Loss Function

As shown in Eq. (28), the weighted sum of generalized IoU (GIoU) loss [42] and binary cross entropy (BCE) loss is exploited to train the proposed tracker. μ and η are weights. y_{init_box} and y_{refine_box} are respectively corresponding bounding box coordinates of key point sets $kp1$ and $kp2$, y_{cls} is predicted classification map. $\hat{y}_{bbox}$ and $\hat{y}_{cls}$ are ground-truth bounding box coordinates and ground-truth classification label, respectively. μ and η are set as 5 and 10 in the training process.

$$Loss_{total} = \mu\left(Loss_{GIoU}(y_{init\_box}, \hat{y}_{bbox}) + Loss_{GIoU}(y_{refine\_box}, \hat{y}_{bbox}) \right) + \eta Loss_{BCE}(y_{cls}, \hat{y}_{cls}) \tag{28}$$

GIoU loss is demonstrated in Eqs. (29) and (30), where $y, \hat{y} \in \mathbb{R}^n$, A^c is the area of the smallest enclosing box between y and $\hat{y}$. BCE loss is calculated in Eq. (31), where y and $\hat{y}$ are respectively the predicted output class probability and ground-truth class label.

$$Loss_{GIoU}(y, \hat{y}) = 1 - GIoU(y, \hat{y}) \tag{29}$$

$$GIoU(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|} - \frac{A^c - |y \cup \hat{y}|}{A^c} \tag{30}$$

$$Loss_{BCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \left[\hat{y}_i \log(y_i) + (1 - \hat{y}_i) \log(1 - y_i) \right] \tag{31}$$

As illustrated in Figure 5, any candidate position (x, y) in the predicted classification map $y_{cls}^{1 \times H_f \times W_f}$ or regression maps $y_{init\_box}^{4 \times H_f \times W_f}$, $y_{refine\_box}^{4 \times H_f \times W_f}$ corresponds to a position (R_x, R_y) of the search region. The relationship χ between the two positions is calculated in Eq. (32). s is the accumulated stride of the tracking framework. H_f, W_f, H and W are respectively the heights and widths of the predicted feature maps and the search region.

$$(R_x, R_y) = \left((x - \frac{W_f}{2}) \cdot s + \frac{W}{2}, (y - \frac{H_f}{2}) \cdot s + \frac{H}{2} \right) \tag{32}$$

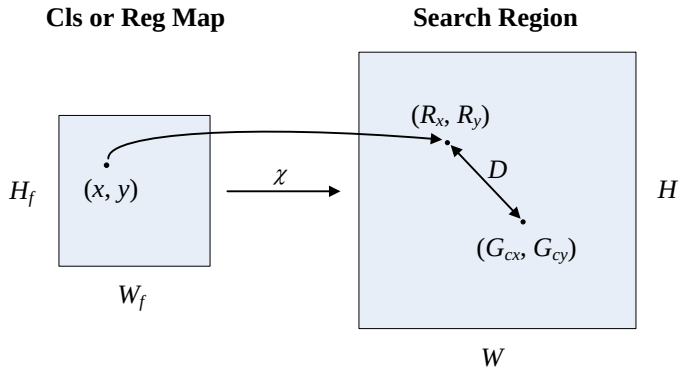


Figure 5 Mapping relation between output map and search region

As shown in Eq. (33), the distances between all $(R_x(i), R_y(j))$ and ground-truth box center (G_{cx}, G_{cy}) are calculated to determine the positive and negative sample positions in the initial regression output y_{init_box}. Positions closest to (G_{cx}, G_{cy}) after mapping are set as positive samples. The remaining positions are set as negative samples. Only positive sample positions are utilized to calculate initial bounding box loss. $\hat{y}_{bbox} \in \mathbb{R}^{4\times H_f \times W_f}$ is the ground-truth label for y_{init_box} and y_{refine_box}, which is obtained by repeating the ground-truth coordinates $[G_{x1}, G_{y1}, G_{x2}, G_{y2}]$ $H_f \times W_f$ times.

$$\min_{\forall i\in[1,W_f]\wedge j\in[1,H_f]} \left\{(R_x(i)-G_{cx})^2+(R_y(j)-G_{cy})^2\right\} \tag{33}$$

As depicted in Eq. (34), the Intersection over Unions ($IoUs$) between initial regression boxes y_{init_box} and ground-truth boxes $\hat{y}_{bbox}$ are calculated to determine the classification label $\hat{y}_{cls}$ and positive and negative samples of classification output y_{cls} and refined regression output y_{refine_box}. $IoUs$ consists of $H_f W_f$ IoU_i, and $IoU_{\max}$ denotes the maximum IoU among $IoUs$. Each IoU_i corresponds to a label l_i, and the relation is demonstrated in Eq. (36). There is at least one label with the value of 1 even if there is no IoU greater than 0.5, which ensures that positive samples can be generated continuously for training. The collection of all labels forms the classification label $\hat{y}_{cls}$. All positive sample positions ($l_i = 1$) and negative sample positions ($l_i = 0$) are employed for calculating classification loss, while only positive sample positions ($l_i = 1$) are utilized to compute the refined bounding box loss for y_{refine_box}.

$$IoUs = \frac{\left|y_{init\_box} \bigcap \hat{y}_{bbox}\right|}{\left|y_{init\_box} \bigcup \hat{y}_{bbox}\right|} \in \mathbb{R}^{H_f \times W_f} \tag{34}$$

$$IoU_{\max} = \max\left\{IoU_i \,\middle|\, IoU_i \in IoUs\right\} \tag{35}$$

$$l_i = \begin{cases} 0, & 0 \le IoU_i < 0.4 \\ 1, & IoU_i \ge 0.5 \\ 1, & IoU_i = IoU_{\max} \end{cases} \tag{36}$$

$$\hat{y}_{cls} = \left\{ l_i \,\middle|\, i \in [1, H_f W_f] \right\} \tag{37}$$

4. Experiments

4.1. Implementation Details

The proposed tracking network is trained on the training sets of 4 publicly available datasets, GOT-10k [22], TrackingNet [23], LaSOT [24] and COCO [43]. The network is validated simultaneously on the validation set of GOT-10k every 10 epochs during training. The backbone network is pre-trained on ImageNet [44] to equip the network with preliminary feature extraction capability. Center jitter and scale jitter are utilized for image augmentation. The center jitter factor and scale jitter factor are respectively set as 3.5 and 0.5 for the search region, and the two factors are set as 0 for the template. The network is trained on a workstation with a Nvidia RTX A6000 GPU and Intel Core i9 CPU. Ubuntu 20.04 is used as the operating system. During training, two frames within the same video in a dataset are sampled and cropped as template and search region, respectively. The template and search region are cropped into square patches with the sizes of 128 pixels and 256 pixels, respectively. The tracking network is trained for 250 epochs with 1000 iterations for one epoch by utilizing AdamW optimizer [45]. The batch size is set as 50. The learning rates of backbone and the remaining parameters are separately set as 10^{-5} and 10^{-4}, which start to decay after 150 epochs with a weight decay factor of 10^{-4}.

4.2. Comparison with State-of-the-art Methods

The proposed tracking method is evaluated and compared with current state-of-the-art tracking algorithms on 5 widely used tracking datasets. The tracking results are shown below. All comparison methods mentioned on 5 datasets utilize a template to assist target localization during tracking.

GOT-10k [22]. GOT-10k dataset contains 560 common objects and 87 motion patterns, which utilizes 10000 and 180 video sequences respectively for training and testing. It provides completely different object categories for training and testing, which evaluates the performance of tracking methods to unknown targets. There are 3 metrics utilized to evaluate tracking performance on GOT-10k, which are respectively *mAO*, $mSR_{0.5}$ and $mSR_{0.75}$. Average overlap (*AO*) measures the average overlap of predicted bounding boxes and ground truth bounding boxes, which is equivalent to Success (AUC). The success rate (*SR*) denotes the percentage of frames with overlaps greater than the threshold 0.5 or 0.75. The mean average overlap (*mAO*) and mean success rate (*mSR*) are two class-balanced metrics.

The proposed tracking algorithm is compared with 16 state-of-the-art tracking methods, respectively. The corresponding results are shown in Table 1. The proposed method TGLC (Ours) outperforms existing tracking approaches on GOT-10k benchmark. The $mSR_{0.75}$ witnesses the largest improvements among them.

Table 1 Tracking results on GOT-10k

Tracking methods	*mAO* ↑	$mSR_{0.5}$ ↑	$mSR_{0.75}$ ↑
TGLC (Ours)	**0.679**	**0.800**	**0.607**
SiamGAT [46]	0.627	0.743	0.488
CMEDFL [47]	0.599	0.682	0.448
Ocean [48]	0.611	0.721	0.473

SiamFC++ [49]	0.595	0.695	0.479
KYS [50]	0.636	0.751	0.515
DCFST [51]	0.638	0.753	0.498
PrDiMP [52]	0.634	0.738	0.543
D3S [53]	0.597	0.676	0.462
DiMP [54]	0.611	0.717	0.492
SiamRPN++ [55]	0.517	0.616	0.325
UTT [56]	0.672	0.763	0.605
[57]	0.425	0.467	0.307
GFS-DCF [58]	0.464	0.482	0.162
AD-LSTM [59]	0.343	0.352	0.109
DTDU [60]	0.375	0.416	0.133
SiamFC [13]	0.348	0.353	0.098

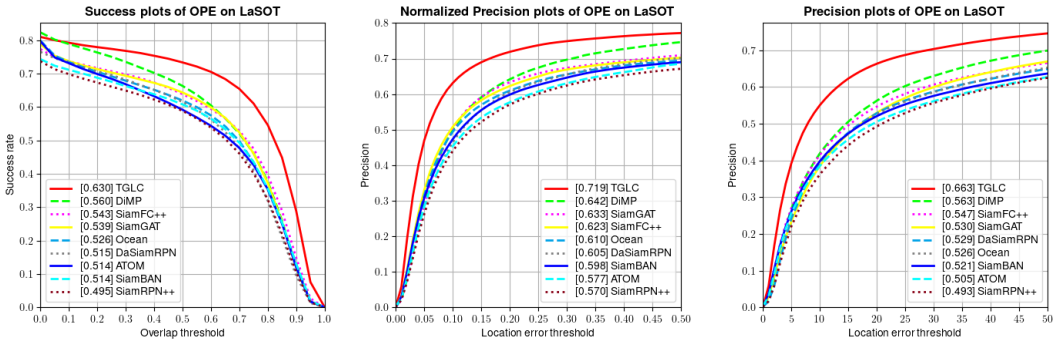
TrackingNet [23]. TrackingNet is currently the largest short-term video target tracking dataset, which consists of 30643 outdoor video sequences. It is divided into 12 sub-training sets and 1 test set. Three evaluation metrics, Precision, Normalized Precision and Success (AUC score) are used to evaluate the tracking performance. The proposed tracking algorithm is compared with 14 state-of-the-art tracking methods on TrackingNet dataset, and the results are illustrated in Table 2. It is obvious that the proposed tracking algorithm TGLC (Ours) surpasses all other tracking methods in terms of 3 evaluation metrics.

Table 2 Tracking performance comparison on TrackingNet

Tracking methods	AUC ↑	Norm Precision ↑	Precision ↑
TGLC (Ours)	**0.808**	**0.859**	**0.790**
UTT [56]	0.797	-	0.770
DualMN [61]	0.778	0.832	0.728
SiamAttn [62]	0.752	0.817	-
TGAN [63]	0.768	0.824	0.710
CMEDFL [47]	0.713	0.761	0.651
SiamFC++ [49]	0.754	0.800	0.705
KYS [50]	0.740	0.800	0.688
DCFST [51]	0.752	0.809	0.700
PrDiMP [52]	0.758	0.816	0.704
CGACD [64]	0.711	0.800	0.693
D3S [53]	0.728	0.768	0.664
DiMP [54]	0.740	0.801	0.687
SiamRPN++ [55]	0.733	0.800	0.694

SiamFC [13]	0.571	0.663	0.533

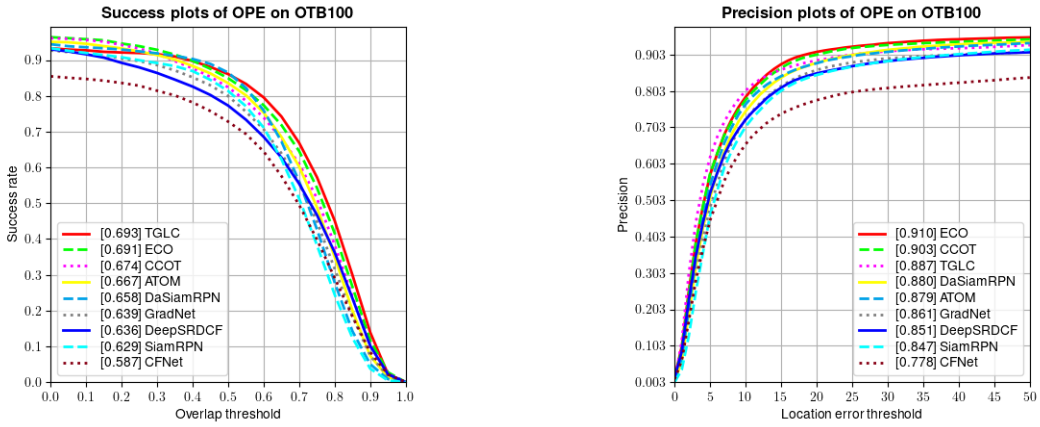
LaSOT [24]. LaSOT is a large-scale long-term tracking benchmark, which consists of two subsets. The main subset contains 1400 video sequences from 70 types of targets. The extended subset has 150 videos from 15 categories of objects. The dataset is divided into a training set and a test set. As shown in Figure 6, the proposed tracking method TGLC (Ours) achieves much better success rate and (normalized) precision than current tracking methods including DiMP [54], SiamFC++ [49], SiamGAT [46], Ocean [48], DaSiamRPN [15], ATOM [65], SiamBAN [66], and SiamRPN++ [55]. This indicates that the proposed approach retains better long-term tracking performance than listed tracking approaches.



(a) Success plot (b) Normalized precision plot (c) Precision plot

Figure 6 Success plot and (normalized) precision plot on LaSOT dataset

OTB100 [25]. OTB100 consists of 100 challenging sequences for visual object tracking. Success rate (AUC) and Precision are utilized to measure the tracking performance of tracking methods, as shown in Figure 7. The proposed method TGLC (Ours) outperforms the existing tracking methods in terms of success rate. Success rate is more important than precision as success rate takes into account scales and centers of predicted boxes simultaneously. The listed methods include ECO [67], CCOT [68], ATOM [65], DaSiamRPN [15], GradNet [69], DeepSRDCF [70], SiamRPN [14] and CFNet [71].

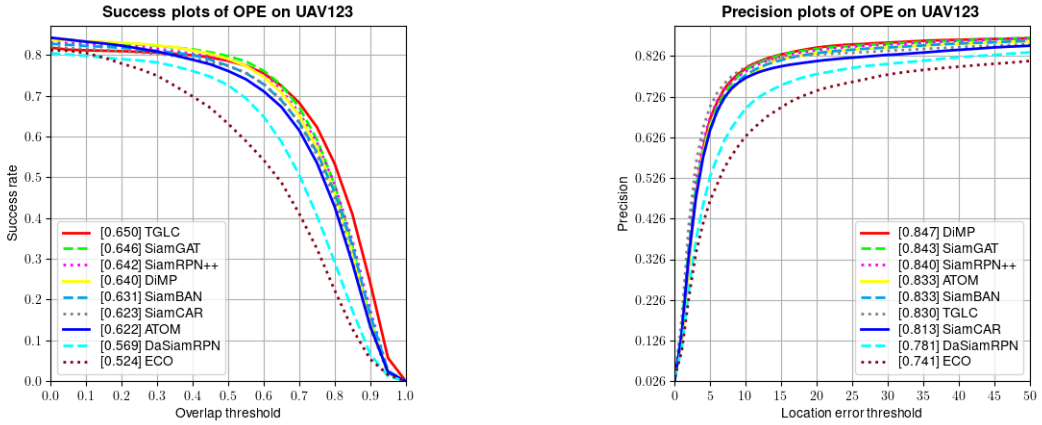


(a) Success plot (b) Precision plot

Figure 7 Success plot and precision plot on OTB100 dataset

UAV123 [26]. UAV123 is a dedicated tracking dataset with videos taken from unmanned aerial

vehicles (UAVs), which is used for evaluating tracking performance by 123 videos captured from drone views. The comparison results are shown in Figure 8. The proposed tracking method TGLC (Ours) achieves a success rate of 0.650, which is competitive and better than listed superior tracking approaches, e.g., SiamGAT [46], SiamRPN++ [55], DiMP [54], SiamBAN [66], SiamCAR [72], ATOM [65], DaSiamRPN [15] and ECO [67]. The precision of TGLC is lower than superior methods, e.g., DiMP. However, precision is less convincing compared to success rate (AUC). Therefore, the more comprehensive AUC is mainly utilized to assess the overall performance of trackers.



(a) Success plot (b) Precision plot

Figure 8 Success plot and precision plot on UAV123 dataset

Seven representative sequences of LaSOT dataset are selected for visualizing the tracking performance of the proposed tracking method TGLC (Ours) and six superior tracking approaches, (DaSiamRPN [15], GlobalTrack [73], SiamBAN [66], Ocean [48], SiamFC++ [49], ATOM [65]), as shown in Figure 9. Sequence bird-17 and person-5 face the problem of background clutter, which denotes that some regions of the background are quite similar to the object. Green and red boxes represent the ground-truth bounding box and the proposed tracking result, respectively. Boxes of other colors denote tracking results of the six existing tracking methods. It is evident that the proposed method has the largest overlap area with the ground truth when other algorithms demonstrate tracking drifts or failures. Partial occlusion occurs in sequence bus-2, and the proposed tracking method can roughly estimate the closest target area with the ground truth even the object is partially occluded. Furthermore, the proposed algorithm displays the maximum similarity with the ground-truth results as well when dealing with other challenges, e.g., illumination variation (guitar-3), out-of-view (horse-1), fast motion (yoyo-7) and scale change and deformation of the object (kangaroo-11). Visualization of the tracking results further confirms the effectiveness of our proposed tracking algorithm TGLC.

Figure 9 Visualization of tracking results on LaSOT

4.3. Ablation Study

A series of ablation experiments are performed to evaluate the effect of different modules on tracking performance. The corresponding results are shown in Table 3 and Table 4.

Backbone ablation. Backbone is used for feature extraction of template and search region, and backbone is a vital module for object tracking. Swin-Tiny [74] and ResNet [37] are respectively utilized to build the backbone. Under the same condition (Backbone + complete

feature fusion module), ResNet50 achieves an AUC of 0.63, outperforming Swin-Tiny (0.595) by 5.9% and ResNet101 (0.623) by 1.1%. Additionally, ResNet50 is the most efficient among three backbone options. ResNet50 is utilized as the final backbone since it achieves the best tracking performance in terms of AUC and tracking speed.

Feature fusion ablation. SE block and CCB block of the proposed feature fusion module are developed to capture the channel information and local feature information of feature maps. In order to explore the effectiveness of the SE block and CCB block, three ablation experiments with or without these two blocks are conducted. An AUC of 0.602 is obtained when removing the SE block and CCB block, which means that pure global information for feature fusion is not able to achieve optimal tracking performance. Gains of 3.3% and 1.3% are achieved when CCB block and SE block are respectively added to the feature fusion module, which proves that global-local information fusion and channel information can substantially improve the tracking performance.

Table 3 Ablation study on LaSOT dataset

Backbones	SE block	CCB block	AUC	Norm Precision
Swin-Tiny	✓	✓	0.595	0.677
ResNet101	✓	✓	0.623	0.709
ResNet50	×	✓	0.622	0.707
ResNet50	×	×	0.602	0.695
ResNet50	✓	✓	**0.630**	**0.719**

Adaptability of CCB block. The proposed CCB block is incorporated with original TransT [19] for further exploring its adaptability and superiority. In order to achieve this, the proposed GLSA and GLCA are separately employed to replace the ECA and CFA modules in TransT [19] for feature fusion. Other components and training details remain the same as the original TransT. The AUC scores of three benchmarks are demonstrated in Table 4.

TransT_N4 and TransT_N2 denote respectively original TransT with 4 and 2 feature fusion layers. TransT_CCB_N4 and TransT_CCB_N2 represent the modified models with 4 and 2 feature fusion layers, separately. It can be seen from the table that the modified models clearly improve the tracking performance on both levels (N2 or N4). Surprisingly, the smaller model TransT_CCB_N2 performs almost on par with TransT_N4 on LaSOT and TrackingNet benchmarks. It further suggests that integration of global local representations and channel information facilitates the feature fusion of two branches. The proposed CCB block can also be integrated into other pure Transformer-based tracking methods to introduce local representations and channel features.

Table 4 Performance comparison between TransT_CCB and TransT at different levels

Methods	LaSOT	TrackingNet	OTB100
TransT_N4 [19]	0.649	0.814	0.694
TransT_CCB_N4 (Ours)	**0.654**	0.820	**0.707**
TransT_N2 [19]	0.642	0.809	0.681
TransT_CCB_N2 (Ours)	0.648	**0.821**	0.684

5. Conclusion

A novel tracking approach named TGLC is proposed in this paper to fully combine global-local feature representations as well as channel information. This end-to-end tracking method is capable of accurately locating the target and predicting immensely appropriate bounding boxes according to the dimension and shape of the target. Experimental results on five popular benchmarks demonstrate the superior tracking performance of the proposed tracking method. Ablation experiments further verify the effectiveness of global-local information aggregation and channel representation modeling for improving tracking performance.

Acknowledgments This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC); and the York Research Chairs (YRC) program.

Data availability All the training and test datasets used in our experiment are public and can be downloaded from their official websites.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Hsu CC, Kang LW, Chen SY, Wang IS, Hong CH, Chang CY (2023) Deep learning-based vehicle trajectory prediction based on generative adversarial network for autonomous driving applications. Multimedia Tools and Applications 82(7):10763-10780

[2] Čegovnik T, Stojmenova K, Tartalja I, Sodnik J (2020) Evaluation of different interface designs for human-machine interaction in vehicles. Multimedia Tools and Applications 79:21361-21388

[3] Tyagi B, Nigam S, Singh R (2022) A review of deep learning techniques for crowd behavior analysis. Archives of Computational Methods in Engineering 29(7):5427-5455

[4] Nigam S, Singh R, Misra AK (2019) A review of computational approaches for human behavior detection. Archives of Computational Methods in Engineering 26:831-863

[5] Singh R, Nigam S, Singh AK, Elhoseny M (2020) Intelligent wavelet based techniques for advanced multimedia applications. Springer International Publishing

[6] Chen Z, Hong Z, Tao D (2015) An experimental survey on correlation filter-based tracking. arXiv preprint arXiv:1509.05520

[7] Nigam S, Khare A (2010) Curvelet transform based object tracking. In: ICCCT. pp 230-235

[8] Nigam S, Khare A (2012) Curvelet transform-based technique for tracking of moving objects. IET Computer Vision 6(3):231-251

[9] Kwak S, Nam W, Han B, Han JH (2011) Learning occlusion with likelihoods for visual tracking. In: ICCV. pp 1551-1558

[10] Vojir T, Noskova J, Matas J (2014) Robust scale-adaptive mean-shift for tracking. Pattern Recognition Letters 49:250-258

[11] Hare S, Golodetz S, Saffari A, Vineet V, Cheng MM, Hicks SL, Torr PH (2015) Struck: Structured output tracking with kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(10):2096-2109

[12] Kalal Z, Mikolajczyk K, Matas J (2011) Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(7):1409-1422

[13] Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH (2016) Fully-convolutional siamese networks for object tracking. In: ECCV. pp 850-865

[14] Li B, Yan J, Wu W, Zhu Z, Hu X (2018) High performance visual tracking with siamese region proposal network. In: CVPR. pp 8971-8980

[15] Zhu Z, Wang Q, Li B, Wu W, Yan J, Hu W (2018) Distractor-aware siamese networks for visual object tracking. In: ECCV. pp 101-117

[16] Wang Q, Zhang L, Bertinetto L, Hu W, Torr PH (2019) Fast online object tracking and segmentation: A unifying approach. In: CVPR. pp 1328-1338

[17] Wang Z, Xu J, Liu L, Zhu F, Shao L (2019) Ranet: Ranking attention network for fast video object segmentation. In: ICCV. pp 3978-3987

[18] Yan B, Zhang X, Wang D, Lu H, Yang X (2021) Alpha-refine: Boosting tracking performance by precise bounding box estimation. In: CVPR. pp 5289-5298

[19] Chen X, Yan B, Zhu J, Wang D, Yang X, Lu H (2021) Transformer tracking. In: CVPR. pp 8126-8135

[20] Yan B, Peng H, Fu J, Wang D, Lu H (2021) Learning spatio-temporal transformer for visual tracking. In: ICCV. pp 10448-10457

[21] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in Neural Information Processing Systems 30

[22] Huang L, Zhao X, Huang K (2019) Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence 43(5):1562-1577

[23] Muller M, Bibi A, Giancola S, Alsubaihi S, Ghanem B (2018) Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: ECCV. pp 300-317

[24] Fan H, Lin L, Yang F, Chu P, Deng G, Yu S, Bai H, Xu Y, Liao C, Ling H (2019) Lasot: A high-quality benchmark for large-scale single object tracking. In: CVPR. pp 5374-5383

[25] Wu Y, Lim J, Yang MH (2015) Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9):1834-1848

[26] Mueller M, Smith N, Ghanem B (2016) A benchmark and simulator for uav tracking. In: ECCV. pp 445-461

[27] Srinivas A, Lin TY, Parmar N, Shlens J, Abbeel P, Vaswani A (2021) Bottleneck transformers for visual recognition. In: CVPR. pp 16519-16529

[28] Xu W, Xu Y, Chang T, Tu Z (2021) Co-scale conv-attentional image transformers. In: ICCV. pp 9981-9990

[29] Dai Z, Liu H, Le QV, Tan M (2021) Coatnet: Marrying convolution and attention for all data sizes. Advances in Neural Information Processing Systems 34:3965-3977

[30] Peng Z, Huang W, Gu S, Xie L, Wang Y, Jiao J, Ye Q (2021) Conformer: Local features coupling global representations for visual recognition. In: ICCV. pp 367-376

[31] Mehta S, Rastegari M (2021) Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178

[32] Woo S, Park J, Lee JY, Kweon IS (2018) Cbam: Convolutional block attention module. In: ECCV. pp 3-19

[33] Hendria WF, Phan QT, Adzaka F, Jeong C (2021) Combining transformer and CNN for object

detection in UAV imagery. ICT Express

[34] Zhang Y, Chen Y, Huang C, Gao M (2019) Object detection network based on feature fusion and attention mechanism. Future Internet 11(1)

[35] Pandey D, Gupta P, Bhattacharya S, Sinha A, Agarwal R (2021) Transformer assisted convolutional network for cell instance segmentation. arXiv preprint arXiv:2110.02270

[36] Petit O, Thome N, Rambour C, Themyr L, Collins T, Soler L (2021) U-net transformer: Self and cross attention for medical image segmentation. In: MLMI. pp 267-276

[37] He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR. pp 770-778

[38] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: ECCV. pp 213-229

[39] Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: CVPR. pp 7132-7141

[40] Ma Z, Wang L, Zhang H, Lu W, Yin J (2020) Rpt: Learning point set representation for siamese visual tracking. In: ECCV. pp 653-665

[41] Yang Z, Liu S, Hu H, Wang L, Lin S (2019) Reppoints: Point set representation for object detection. In: ICCV. pp 9657-9666

[42] Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S (2019) Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR. pp 658-666

[43] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: ECCV. pp 740-755

[44] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC (2015) Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115(3):211-252

[45] Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101

[46] Guo D, Shao Y, Cui Y, Wang Z, Zhang L, Shen C (2021) Graph attention tracking. In: CVPR. pp 9543-9552

[47] Zhu P, Yu H, Zhang K, Wang Y, Zhao S, Wang L, Zhang T, Hu Q (2021) Learning dynamic compact memory embedding for deformable visual object tracking. arXiv preprint arXiv:2111.11625

[48] Zhang Z, Peng H, Fu J, Li B, Hu W (2020) Ocean: Object-aware anchor-free tracking. In: ECCV. pp 771-787

[49] Xu Y, Wang Z, Li Z, Yuan Y, Yu G (2020) Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In: AAAI 34(07):12549-12556

[50] Bhat G, Danelljan M, Gool LV, Timofte R (2020) Know your surroundings: Exploiting scene information for object tracking. In: ECCV. pp 205-221

[51] Zheng L, Tang M, Chen Y, Wang J, Lu H (2020) Learning feature embeddings for discriminant model based tracking. In: ECCV. pp 759-775

[52] Danelljan M, Gool LV, Timofte R (2020) Probabilistic regression for visual tracking. In: CVPR. pp 7183-7192

[53] Lukezic A, Matas J, Kristan M (2020) D3s-a discriminative single shot segmentation tracker.

In: CVPR. pp 7133-7142

[54] Bhat G, Danelljan M, Gool LV, Timofte R (2019) Learning discriminative model prediction for tracking. In: ICCV. pp 6182-6191

[55] Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J (2019) Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: CVPR. pp 4282-4291

[56] Ma F, Shou MZ, Zhu L, Fan H, Xu Y, Yang Y, Yan Z (2022) Unified transformer tracker for object tracking. In: CVPR. pp 8781-8790

[57] Zhang H, Zhang Z, Zhang J, Zhao Y, Gao M (2023) Online bionic visual siamese tracking based on mixed time-event triggering mechanism. Multimedia Tools and Applications 82(10):15199-15222

[58] Javed S, Mahmood A, Ullah I, Bouwmans T, Khonji M, Dias JMM, Werghi N (2022) A novel algorithm based on a common subspace fusion for visual object tracking. IEEE Access 10:24690-24703

[59] Zhang H, Liang J, Zhang J, Zhang T, Lin Y, Wang Y (2023) Attention-driven memory network for online visual tracking. IEEE Transactions on Neural Networks and Learning Systems

[60] Liu J, Wang Y, Huang X, Su Y (2022) Tracking by dynamic template: Dual update mechanism. Journal of Visual Communication and Image Representation 84:103456

[61] Wang J, Zhang H, Zhang J, Miao M, Zhang J (2022) Dual-branch memory network for visual object tracking. In: PRCV. pp 646-658

[62] Yu Y, Xiong Y, Huang W, Scott MR (2020) Deformable siamese attention networks for visual object tracking. In: CVPR. pp 6728-6737

[63] Yang K, Zhang H, Zhou D, Liu L (2021) TGAN: A simple model update strategy for visual tracking via template-guidance attention network. Neural Networks 144:61-74

[64] Du F, Liu P, Zhao W, Tang X (2020) Correlation-guided attention for corner detection based visual tracking. In: CVPR. pp 6836-6845

[65] Danelljan M, Bhat G, Khan FS, Felsberg M (2019) Atom: Accurate tracking by overlap maximization. In: CVPR. pp 4660-4669

[66] Chen Z, Zhong B, Li G, Zhang S, Ji R (2020) Siamese box adaptive network for visual tracking. In: CVPR. pp 6668-6677

[67] Danelljan M, Bhat G, Shahbaz Khan F, Felsberg M (2017) Eco: Efficient convolution operators for tracking. In: CVPR. pp 6638-6646

[68] Danelljan M, Robinson A, Shahbaz Khan F, Felsberg M (2016) Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: ECCV. pp 472-488

[69] Li P, Chen B, Ouyang W, Wang D, Yang X, Lu H (2019) Gradnet: Gradient-guided network for visual object tracking. In: ICCV. pp 6162-6171

[70] Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2015) Convolutional features for correlation filter based visual tracking. In: ICCV. pp 58-66

[71] Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PH (2017) End-to-end representation learning for correlation filter based tracking. In: CVPR. pp 2805-2813

[72] Guo D, Wang J, Cui Y, Wang Z, Chen S (2020) SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In: CVPR. pp 6269-6277

[73] Huang L, Zhao X, Huang K (2020) Globaltrack: A simple and strong baseline for long-term

tracking. In: AAAI 34(07):11037-11044

[74] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp 10012-10022