Multimedia Systems (2024) 30:245
https://doi.org/10.1007/s00530-024-01423-8

REGULAR PAPER

Propagating prior information with transformer for robust visual object tracking

Yue Wu[1] · Chengtao Cai[1,2] · Chai Kiat Yeo[3]

Received: 31 January 2024 / Accepted: 15 July 2024

Abstract

In recent years, the domain of visual object tracking has witnessed considerable advancements with the advent of deep learning methodologies. Siamese-based trackers have been pivotal, establishing a new architecture with a weight-shared backbone. With the inclusion of the transformer, attention mechanism has been exploited to enhance the feature discriminability across successive frames. However, the limited adaptability of many existing trackers to the different tracking scenarios has led to inaccurate target localization. To effectively solve this issue, in this paper, we have integrated a siamese network with the transformer, where the former utilizes ResNet50 as the backbone network to extract the target features, while the latter consists of an encoder and a decoder, where the encoder can effectively utilize global contextual information to obtain the discriminative features. Simultaneously, we employ the decoder to propagate prior information related to the target, which enables the tracker to successfully locate the target in a variety of environments, enhancing the stability and robustness of the tracker. Extensive experiments on four major public datasets, OTB100, UAV123, GOT10k and LaSOText demonstrate the effectiveness of the proposed method. Its performance surpasses many state-of-the-art trackers. Additionally, the proposed tracker can achieve a tracking speed of 60 fps, meeting the requirements for real-time tracking.

Keywords Visual object tracking · Siamese network · Transformer · Prior information

1 Introduction

In the ever-evolving field of computer vision, video understanding tasks [1, 2] have been receiving increasing attention. Efforts are being made to improve the techniques for object tracking [3–5], video classification [6], and action recognition [7] to meet the growing demand for effective video analysis. Among them, single-object tracking (SOT) has emerged as a pivotal area of research, driven by its extensive applications in autonomous vehicles, surveillance, human–computer interaction and augmented reality. The essence of single-object tracking lies in the continuous localization of a target within a video sequence, where the target is initially labeled in the first frame. Challenges such as occlusions, deformation and varying lighting conditions have historically hindered tracking performance. Therefore, designing a robust and stable tracker is of great importance.

Before deep learning became widespread, correlation filters were crucial in visual object tracking. Staple [8] combines color information with traditional filter responses, while ECO [9] integrates deep feature extraction with correlation filters, reducing complexity but maintaining high performance. With the robust emergence of deep learning, MDNet [10] utilizes shallow CNNs for multi-domain learning. The Siamese network, due to its weight-sharing characteristic, became popular in tracking [11, 12], with SiamFC [13] being the first to use it, employing AlexNet [14] for feature extraction. SiamRPN's

Communicated by Junyu Gao.

✉ Chengtao Cai
caichengtao@hrbeu.edu.cn

Yue Wu
wuyue1116@hrbeu.edu.cn

Chai Kiat Yeo
asckyeo@ntu.edu.sg

[1] School of Intelligent Science and Engineering, Harbin Engineering University, Harbin 150001, China

[2] Key laboratory of Intelligent Technology and Application of Marine Equipment, Ministry of Education, Harbin Engineering University, Harbin 150001, China

[3] School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore

Published online: 13 August 2024

family [15–17] introduce the RPN [18], transforming tracking into a classification-regression problem. Due to the complexity and time-consuming tuning of anchor-based parameters in RPN, anchor-free strategies were adopted [19–22] to simplify the model.

In recent years, the transformer [23], successful in natural language processing, have been adopted in computer vision [24]. With attention mechanisms, they help networks focus on the useful information. DETR [25] pioneers this by using CNNs to extract and transformers to enhance features, setting a new direction. Deformable DETR [26] improves on this with a deformable attention mechanism to address slow convergence and limited feature resolution. In visual object tracking, HiFT [27] and TCTrack [28] incorporate the transformer, but with post cross-correlation [13] operation before it. This process is a local linear matching method that will result in the loss of semantic information and has a tendency to fall into local optima. Recognizing the shortcomings of cross-correlation, most transformer-based trackers [29–36] have replaced it with the transformer. Among them, TransT [29] uses a variant of the transformer. The extracted template and search features are fed into a combination of an Ego-Context Augment (ECA) and a Cross-Feature Augment (CFA) module, respectively, with one combination acting as the encoder and the other as the decoder, followed by another CFA module to fuse the two features. HCAT [30], building on TransT, removes the two ECA modules. TrTr [36] uses the traditional transformer, with the template features as the encoder input and the search features as the decoder input. In this paper, we introduce a novel tracker that propagates prior information using the transformer for visual object tracking (PI-Trans) through a combination of a siamese network and the transformer. The former utilizes ResNet50 as the backbone network, extracting features from both the template and the search images. Different from [29, 30, 36], these features are flattened and concatenated before being fed into the encoder of the transformer. In the decoder of the transformer, we utilize the flattened template features as the input to acquire the prior information relevant to the target. The obtained information will be propagated with the encoder's output through the process of information interaction, enabling the tracker to rapidly adapt to the various scenes and effectively differentiate the targets from the distractions. In summary, PI-Trans concatenates template and search features in the encoder for deep interaction. The decoder uses the reliable features, which come from the unchanged template image rather than the dynamic search image. Meanwhile, the prediction network consists of two branches, namely the classification and the regression branches, with the former predicting the foreground/background while the latter predicts the bounding boxes without any anchor-based parameters. making the model more concise. Our contributions are summarized as follows:

(1) The designed tracker, PI-Trans, is an end-to-end tracker, comprising a siamese network, a transformer and a prediction network. It discards the anchor-based strategy, achieving a speed of 60 fps during tracking, fulfilling the requirements for real-time applications.
(2) We employ the encoder of the transformer to fully exploit the global contextual information and integrate target-related prior information into the decoder of the transformer, enabling the tracker to adapt to different tracking scenarios and accurately distinguishing between the distractors and the targets.
(3) To prove the effectiveness of PI-Trans, we test it on four major public tracking datasets: OTB100 [37], UAV123 [38], GOT10k [39] and LaSOText [40] as well as comparing it against numerous state-of-the-art (SOTA) trackers. The experimental results show that PI-Trans has excellent performance and is in a leading position.

The paper consists of five sections. The first section is the introduction; the second reviews the related works in visual object tracking; the third elaborates our proposed methods in details and the fourth details the extensive experiments conducted for validation. The last section concludes the paper.

2 Related works

2.1 Deep learning in tracking

Deep learning has rapidly evolved in recent years, and in the field of visual object tracking, methods based on deep learning can automatically extract and learn the most relevant features for tracking. This has made the tracking process more efficient and effective. DiMP [41] employs an end-to-end training approach with online updates for better handling of objects not in the training set. PrDiMP [42] introduces the probabilistic regression formulation, which provides clearer, probabilistic insights to enhance tracking accuracy. Another classical method is based on the siamese network. In such networks, the two parallel branches share the same backbone network and weights. The upper branch is commonly referred to as the template branch and the lower one is known as the search branch. SiamFC [13] uses cross-correlation operations to fuse the template and search features. This helps in identifying the location of the target in the search image by finding the area with the highest similarity to the template. Subsequently, the SiamRPN [15] series make a significant splash in the tracking field, utilizing an anchor strategy to select the appropriate predicted bounding boxes. Among them, SiamRPN is the earliest. DaSiamRPN [16] and SiamRPN++ [17] are its improved versions. DaSiamRPN introduces the concept of being distractor-aware and expands the training dataset. On the other hand, SiamRPN++ not only introduces a deeper backbone network structure Resnet [43] but also improves on the cross-correlation by proposing a depth-wise cross-correlation. Later, anchor-free trackers, such as, SiamBAN [19] utilizes a box adaptive head to obtain the predicted bounding boxes while SiamFC++ [20] is an advanced version of SiamFC which employs target size estimation method. SiamCAR [21] and SiamGAT [22] incorporate a centerness branch in the classification branch, a concept introduced in FCOS [44]. The inclusion of centerness branch helps to eliminate bounding boxes that are far from the target center. The above-mentioned Siamese-based trackers, except for SiamGAT, use cross-correlation to merge the template and search features. This method convolves template features with search features, leading to local linear matching that cannot make full use of the global contextual information and easily fall into local optima. Although SiamGAT replaces cross-correlation with a graph attention method, it still struggles to accurately distinguish between the targets and distractors in some tracking scenarios.

2.2 Transformer in tracking

The transformer [23], with its exceptional global modeling capabilities, has achieved great success in object tracking tasks. Transformer is primarily composed of two main modules: an encoder and a decoder, both of which are based on the attention mechanisms to enhance the representational ability of the model. HiFT [27] introduces hierarchical feature learning, allowing for interactive fusion between shallow and deep layers, while TCTrack [28] integrates a TAdaCNN [45] with the transformer to explore temporal contexts. It is important to note that both trackers add the transformer after performing cross-correlation operation. As this approach has its limitations and prevents the transformer from realizing its full potential, these transformer-based trackers [29–36] use the transformer to replace the cross-correlation operation. Among them, TransT [29] designs two modules: Ego-Context Augment (ECA) and Cross-Feature Augment (CFA). The template and search features first pass through an ECA and a CFA module, respectively, and finally, another CFA module is used for feature fusion. This undoubtedly increases the computational load of the model. Different from TransT, HCAT [30] removes the ECA module both in the template and search branches, but adds a feature sparsification module in the former. MTFM [34] also uses ECA and CFA modules. BANDT [31] introduces the deformable transformer with a border-aware network, while IoUformer [35] designs a IoU predictor to generate reliable IoU values. DTT [32] and TrTr [36] use the traditional transformer to improve the tracking performance. At present, it is still necessary to explore more potential of the transformer in the field of visual object tracking. In this paper, we also discard the cross-correlation operation. For the template and search features, we cascade them and then directly input them into the encoder of the transformer for global contextual information exploration without any extra fusion modules, thereby simplifying the model and making it more streamlined and efficient. We further integrate target-related prior information into the decoder to build a robust tracker.

3 Methodology

We propose a novel tracker called PI-trans, which utilizes the transformer to propagate prior information, allowing the tracker to adapt to various tracking scenarios. We first introduce the backbone network within the siamese network, then present the detailed design of the transformer, and finally the composition of the prediction network.

3.1 Backbone network

As shown in Fig. 1, the input of the backbone network consists of a pair of images, i.e. a template image and a search

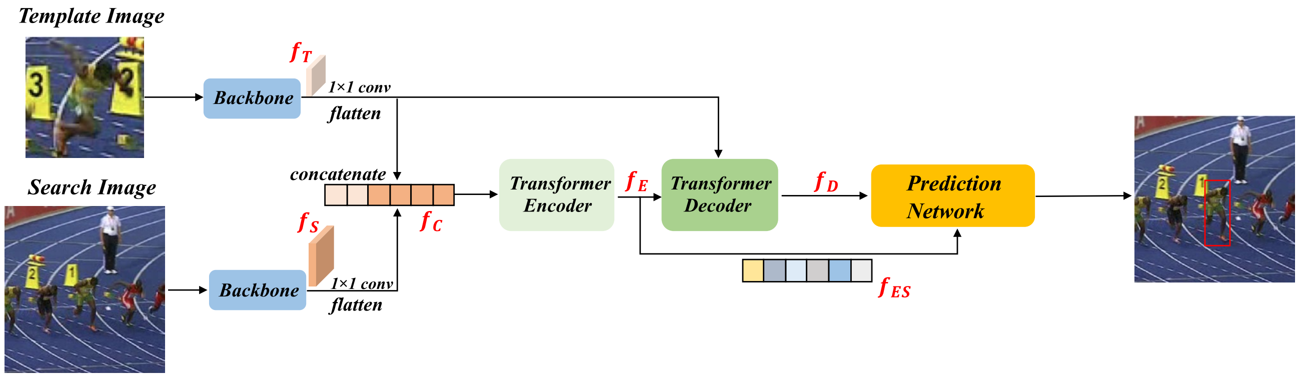

Fig. 1 Overview of PI-Trans. On the left is the backbone network ResNet50 followed by the transformer and then the prediction network

image. The template image is a cropped version of the first frame of a video sequence, centered around the tracking target and its surrounding environment. It is important to note that the template remains unchanged throughout the tracking process. The search image, centered on the target from the previous frame, encompasses the potential movement range of the target and is typically larger than the template image. Both the template and search images are fed into a modified ResNet50 [43] for feature extraction, yielding template feature f_T and search feature f_S. The modifications to the backbone network include removing its last stage and utilizing the output from the fourth stage to enlarge the resolution of the extracted features. Furthermore, the 3×3 convolution in this stage is modified into a dilated convolution with a stride of 2, effectively widening the receptive field.

3.2 Transformer

After the features are extracted, as shown in Fig. 1, we first use two 1×1 convolutions to reduce the channel dimension of the template and search features, separately. Then, we flatten and concatenate them along the spatial dimension to obtain the new features f_C. These features are then fed into the encoder of the transformer. Figure 2 shows the basic components of the encoder and decoder, which contain multi-head attention, add & norm, and a feed forward network. The multi-head cross-attention module in the decoder has the function of information interaction. The multi-head attention module is composed of h heads and we set h to 8. Its role is to focus on the different aspects of the input information.

The attention function can be described as:

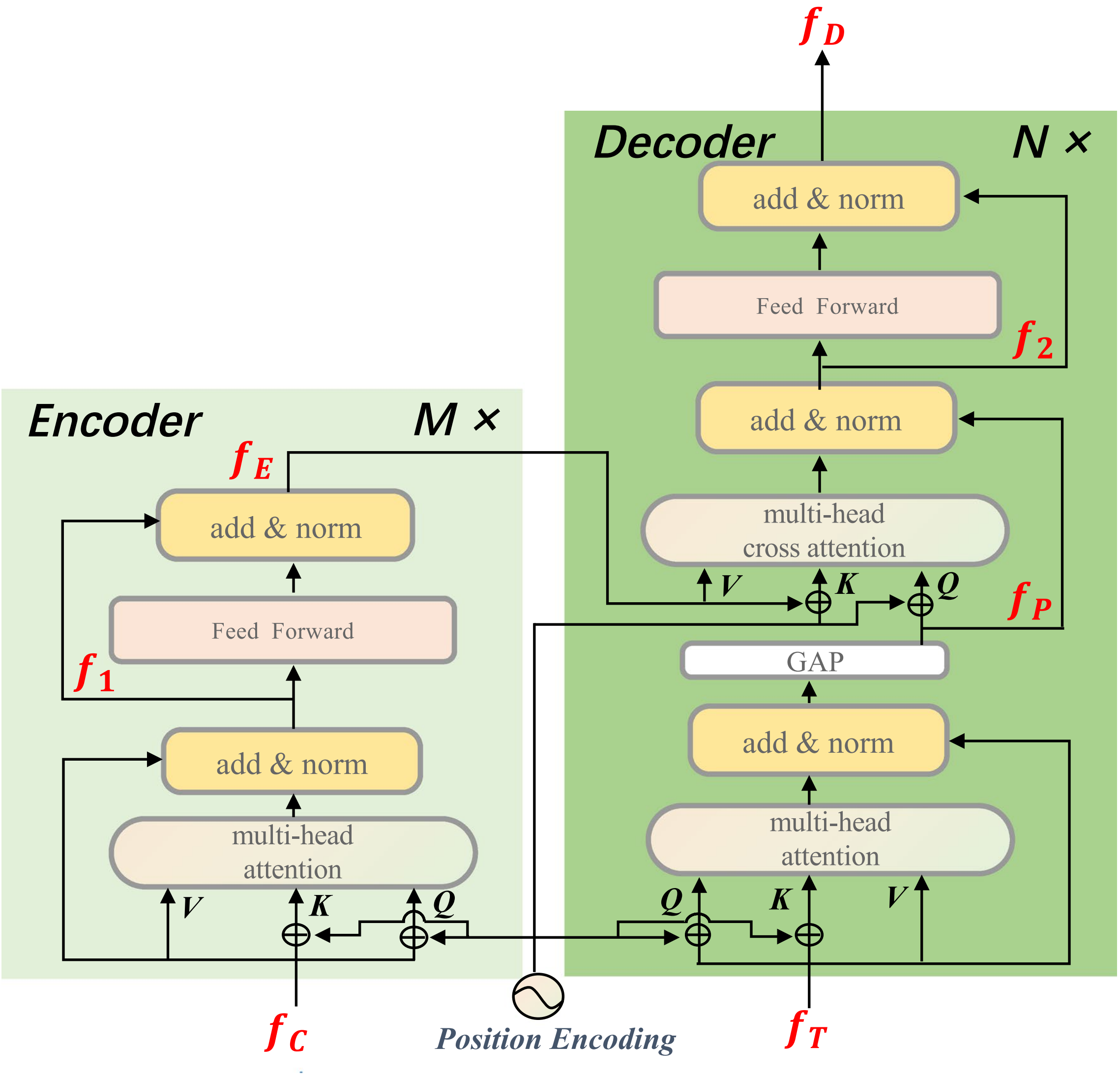

Fig. 2 Architecture of the transformer

$$Attention(Q, K, V) = Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{D_k}}\right)V \tag{1}$$

Here, Q represents the query, K represents the key and V represents the value. $\sqrt{D_k}$ serves as a scale factor to prevent the vanishing of gradients. A single head's attention can be represented as follows:

$$H_j = Attention(QW_j^Q, KW_j^K, VW_j^V) \tag{2}$$

where, $W_j^Q \in \mathbb{R}^{d \times D_k}$, $W_j^K \in \mathbb{R}^{d \times D_k}$ and $W_j^V \in \mathbb{R}^{d \times D_v}$. We set d = 256, $D_k = D_v$ = d/h = 32. Finally, multi-head attention is the concatenation of all the single heads.

$$multihead(Q, K, V) = Cat(H_1, H_2, \dots H_h)W^O \tag{3}$$

Here, $W^O \in \mathbb{R}^{d \times d}$. W_j^Q, W_j^K, W_j^V and W^O are matrices with learnable parameters. Norm in the add & norm denotes the layer normalization and the purpose of the feed forward network is to enhance the adaptability of the model. From Fig. 2, we summarize the process of obtaining the output f_E of the encoder as follows:

$$\begin{aligned} f_1 &= norm(f_C + mhattn(f_C + pos, f_C + pos, f_C)), \\ f_E &= norm(f_1 + ffn(f_1)) \end{aligned} \tag{4}$$

Here, pos denotes position encoding and mhattn denotes multihead attention. Due to the insensitivity of the attention mechanism to the position information, it cannot comprehend the input sequence order. Therefore, we add sinusoidal position embeddings to each Q and K in both the encoder and decoder (see Fig. 2), treating them as position encoding. This approach is similar to the position encoding used in DETR [25]. By capturing the inter-dependencies between all elements within the input features, the encoder effectively leverages the global contextual information to enhance the original features. It provides the tracker with discriminative features that are critical to accurately locate the target.

For the decoder, its first input is the template feature f_T, since the template image is the first cropped frame of the video sequence, which is centered on the tracked target and its surroundings. Therefore, it is accurate and reliable to utilize the template features to obtain the prior information. First, f_T goes through a multi-attention and add & norm module, and then a global average pooling (GAP) module is used to further integrate the information to obtain f_P. This removes the distracting information that is not related to the target, thus allowing the tracker to focus on the target and be able to adapt to different tracking scenarios. We consider f_P as the target-related prior information, which will be propagated in the next step of information interaction with the encoder's output. We use mhattn to denote multihead cross attention and summarize the decoder's operation as follows:

$$\begin{aligned} f_P &= GAP(norm(f_T + mhattn(f_T + pos, f_T + pos, f_T))), \\ f_2 &= norm(f_P + mhcattn(f_P + pos, f_E + pos, f_E)), \\ f_D &= norm(f_2 + ffn(f_2)) \end{aligned} \tag{5}$$

3.3 Prediction network

Figure 3 shows the details of the designed prediction network. We first extract the search features f_{ES} from the output of the encoder. These are then channel-wise multiplied with the output f_D of the decoder. Since the decoder incorporates target-related prior information, the resulting features f_3 become highly sensitive to the target's location. Following this, an element-wise sum is carried out between f_3 and f_{ES} to enrich the information. Finally, the resulting features f_4 are fed into the classification-regression network to generate the prediction bounding boxes with its associated classification scores.

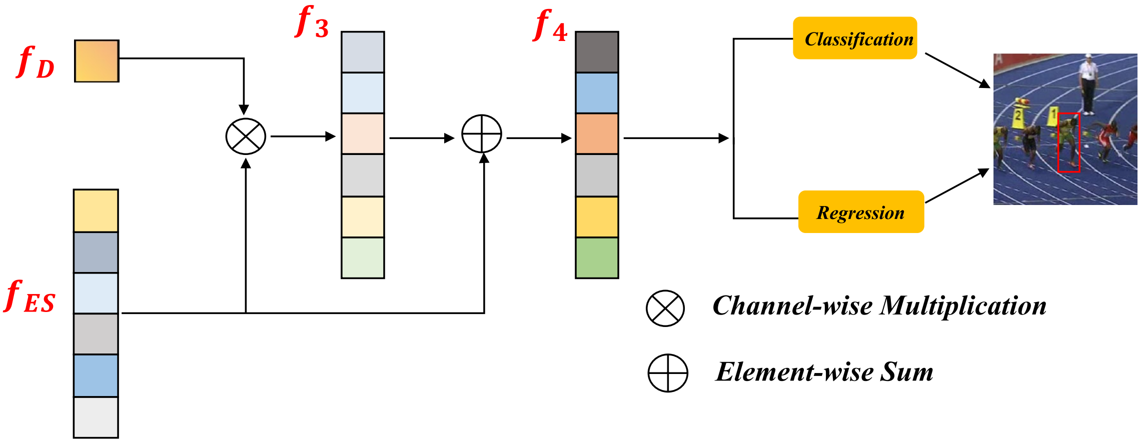

Fig. 3 Architecture of the prediction network

The classification and regression networks, each contains a three-layer multilayer perceptron, employing RELU [46] as the activation function. In the training phase, for the classification loss, we use focal loss [47] and it is defined as:

$$L_{cls} = \frac{1}{N_p} \sum_{(i,j)} [\beta_1 L_{focal}(c(i,j), c^*(i,j))] \tag{6}$$

Here, β_1 is the balance parameter and is set to 1, c(i,j) denotes the predicted classification score. When (i,j) is a positive sample, $c^*(i,j)$ equals 1, otherwise, $c^*(i,j)$ equals 0. For (i,j), if it is within the ground-truth bounding box, we define it as a positive sample; otherwise, it is considered a negative sample. N_p denotes the count of positive samples.

For the regression loss, we use DIOU loss [48] and L1 loss. The former can decrease the distance between the predicted bounding boxes and ground-truth bounding boxes, thereby acquiring high-quality predicted bounding boxes. The regression loss can be formulated as:

$$L_{reg} = \frac{1}{N_p} \sum_{(i,j)} 1_{\{c^*(i,j)=1\}} [\beta_2 L_{DIOU}(b(i,j), b^*(i,j)) + \beta_3 L_1(b(i,j), b^*(i,j))] \tag{7}$$

Here, the balance parameters β_2 and β_3 are set to 2 and 3 respectively. Since the regression branch generates the width, height and center coordinates of the predicted box without any anchor-based parameters, it renders the model more streamlined and efficient. Table 1 shows the annotated notations used in Figs. 1, 2 and 3.

4 Experiments

4.1 Implementation details

PI-Trans is implemented in Python using the PyTorch 1.5.1. It is trained on two 24GB NVIDIA RTX 3090 GPUs and tested on a single NVIDIA RTX 3090 GPU. We train our model on COCO [49], ImageNet-VID [50], ImageNet-DET [50], GOT-10k [39], and LaSOT [51], employing a series of data augmentation methods, such as level flipping and luminance jitter to further extend the training set. The template image is sized at 128 × 128, while the search image is 256 × 256. The backbone network is initialized with parameters from a pre-trained ResNet50 [43] on ImageNet [50]. We use AdamW [52] as our optimizer. The learning rate for the backbone network is set to 0.00001, and 0.0001 is used for the other parameters, with a weight decay of 0.0001. The model undergoes a total of 500 training epochs, with the learning rate dropping by a factor of 10 after 400 epochs. M and N in Fig. 2 denote the layers used in the encoder and decoder respectively, and we set M = N = 6.

Table 1 Annotated notations used in the Figs. 1, 2 and 3

Notations	
f_T	Template features
f_S	Search features
f_C	Cascaded template and search features
f_1	Encoder's multi-head attention and add & norm output
f_E	Encoder's output
f_P	Prior information
f_2	Decoder's second multi-head attention and add & norm output
f_D	Decoder's output
f_{ES}	Search features extracted from the encoder's output
f_3	The output of channel-wise multiplication of f_D and f_{ES}
f_4	The output of element-wise sum of f_3 and f_{ES}

In the tracking phase, the template image, being the first frame of the video sequence, stays constant and only the search image is updated. The prediction network outputs the bounding boxes and classification scores, where each bounding box is associated with a corresponding classification score. In order to minimize the boundary effects, the cosine window is applied to the classification scores to obtain the final scores. The result is the box with the highest final score. We test our tracker on OTB100 [37], UAV123 [38], GOT10k [39] and LaSOText [40] datasets and compare it against existing SOTA methods.

4.2 OTB100

OTB100 [37] consists of 100 video sequences that encompass various real-world tracking scenarios, such as, illumination variations (IV), occlusion (OCC), scale variation (SV) and background clutter (BC). OTB100 has two evaluation metrics, namely, success and precision. Success measures the IOU between the predicted and ground-truth bounding boxes, serving as a primary metric for ranking trackers. Its plot displays the proportion of frames where the overlap rate surpasses the given threshold. Precision measures the distance between the centers of the predicted and ground-truth bounding boxes. Its plot displays the proportion of frames where the predicted location is within 20 pixels from the ground-truth. OTB100 uses OPE (one-pass-evaluation) for tracker evaluation.

In Figs. 4 and 5, we compare the overall performance of PI-Trans, HCAT [30], TransT [29], HiFT [27], TCTrack [28], SiamFC++ [20], DaSiamRPN [16], SiamRPN [15], SiamFC [13] and Staple [8], as well as their performance under the BC, IV, OCC and DEF attributes. In terms of overall performance, PI-Trans achieves 69.5% success and 90.8% precision, which is 1.5% higher in success compared to the second-place transformer-based tracker HCAT, and 1.8% higher in precision compared to the

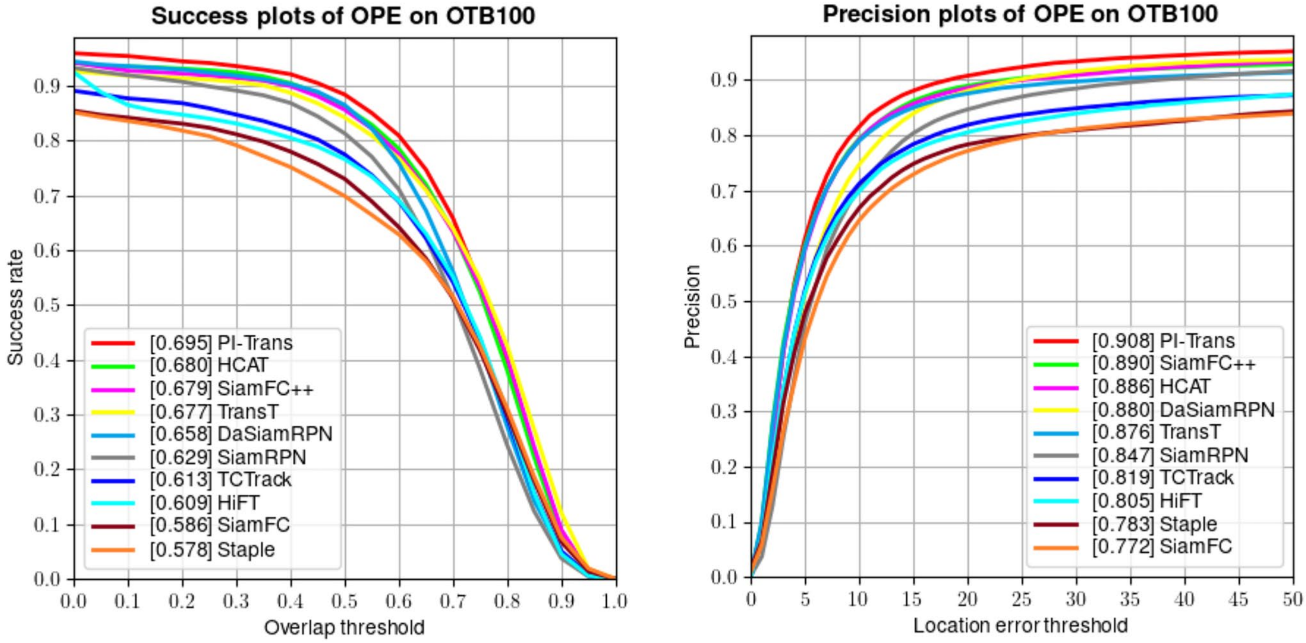

Fig. 4 Performance of PI-Trans and existing SOTA trackers on OTB100 in terms of Success and Precision. It is best viewed when zoomed in

second-place siamese-based tracker SiamFC++. Pertaining to the performance across the different attributes, PI-Trans still maintains a leading position, proving that it not only possesses better performance than many SOTA trackers, but also has strong adaptability to the various tracking scenarios.

4.3 UAV123

UAV123 [38] is designed for testing trackers in unmanned aerial vehicle (UAV) scenarios. It comprises 123 testing videos, with a total frame count exceeding 100k. Given that UAV-captured targets are typically smaller, testing on it is much more challenging than on OTB100 for trackers with poor discriminative ability. UAV123 employs the same evaluation metrics as OTB100, namely, success and precision and also uses OPE to evaluate the trackers.

Figure 6 shows the performance of PI-Trans against SiamGAT [22], HCAT [30], SiamCAR [21], SiamRPN++ [17], TCTrack [28], SiamBAN [19], HiFT [27], DaSiamRPN [16] and ECO [9]. It can be seen that PI-Trans achieves a success rate of 65.6% and a precision of 85.5%, which is respectively 1% and 1.2% higher than the second-place SiamGAT, and 2.9% and 4.3% higher than the third-placed HCAT. This indicates that even in the tracking of small targets, PI-Trans exhibits excellent discriminative abilities, effectively distinguishing between the targets and distractors, thereby once again proving its strong adaptability. Table 2 further shows the success rate of PI-Trans compared to three transformer-based trackers [27, 28, 30] and one siamese-based tracker [17] on the BC (background clutter), IV (illumination variation) and POC (partial occlusion) attributes in the UAV123 dataset, where it can be clearly seen that PI-Trans is at the forefront.

4.4 GOT10k

GOT10k [39] is a large-scale dataset, with a wide variety of objects and scenarios. it includes 563 object classes and covers 87 object sub-classes and is known for its diversity. For fair comparison, as per the official website of GOT-10k, a model is required to be trained exclusively on its training set and then tested on its test set. GOT-10k includes 180 test videos. Its evaluation metrics are AO (Average Overlap), SR_0.5 (Success Rate), and SR_0.75. AO measures the mean IOU between the predicted and ground-truth boxes across all frames and is the primary metric to rank the trackers. SR calculates the percentage of frames where the IOU exceeds a set threshold, i.e, 0.5 and 0.75.

We compare PI-Trans with several SOTA trackers, including Staple [8], MDNet [10], ECO [9], SiamFC [13], SiamRPN [15], SiamRPN++ [17], SiamCAR [21], SiamFC++ [20], DiMP [41], PrDiMP [42], SiamGAT [22], TransT [29], HCAT [30], BANDT [31], DTT [32], TT-DiMP [33], MTFM [34] and IoUformer [35] in Table 3, TransT achieves the best AO score at 67.1%, our proposed tracker, PI-Trans, achieved AO, $SR_{0.5}$, and $SR_{0.75}$ scores, at 66.2%, 77.0%, and 58.3%, respectively. These scores are 1.1%, 0.5%, and 1.6% higher than HCAT. This demonstrates PI-Trans's adaptability to the different

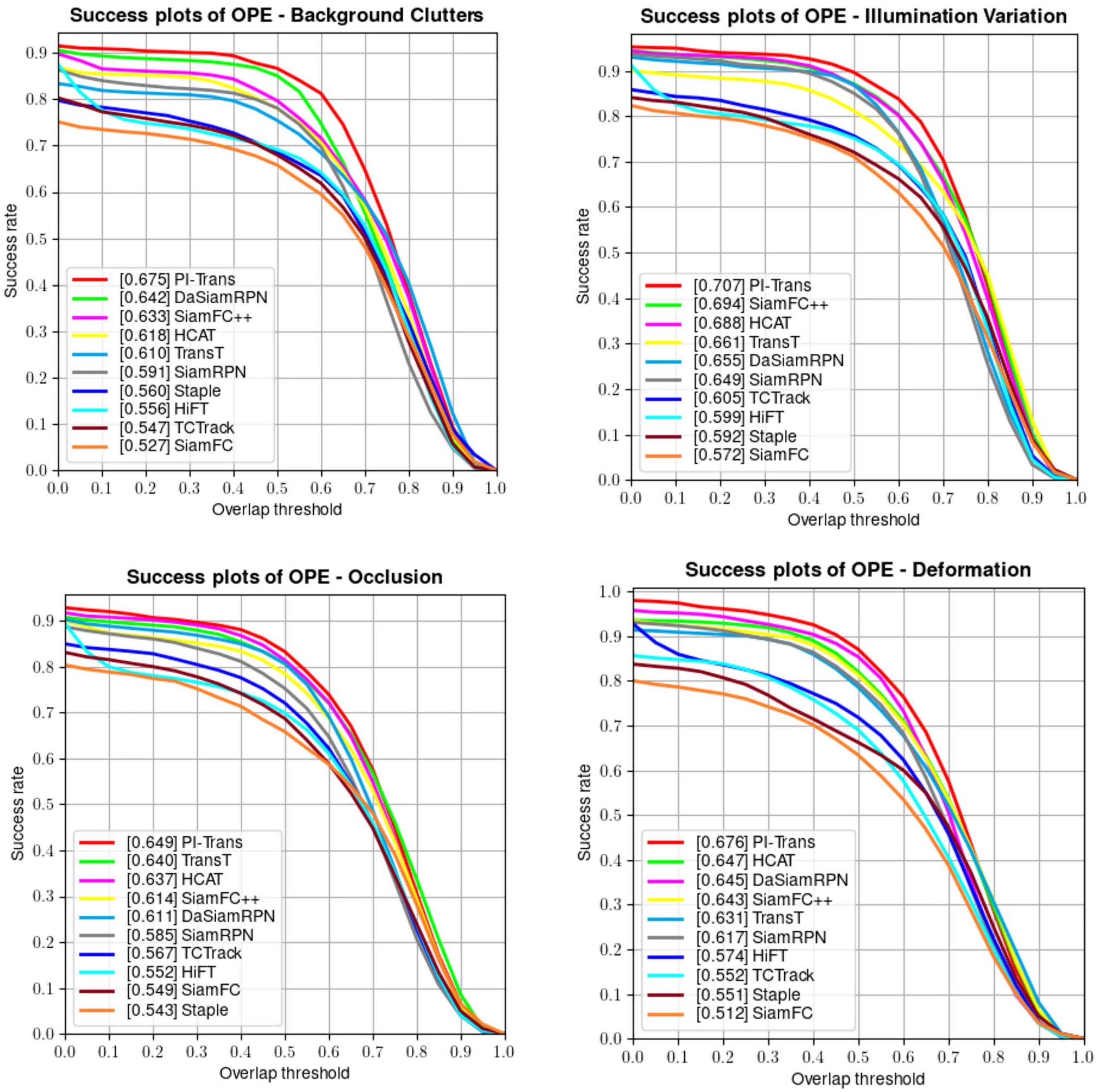

Fig. 5 Comparison of Success for BC, IV, OCC and DEF attributes on OTB100. It is best viewed when zoomed in

tracking scenarios and proves its effectiveness.The best results are highlighted in bold in Table 3.

4.5 LaSOText

LaSOText [40] is a large-scale and long-term dataset composed of 150 test videos and encompasses 15 target categories. It is an extended version of LaSOT, featuring more distractors similar to the targets in its video sequences. Due to its long-term tracking nature, a tracker without strong discriminative and generalization capabilities will easily lose the target in these challenging scenarios, leading to bad tracking performance. Unlike OTB100 and UAV123, LaSOText includes an additional evaluation metric, normalized precision, which we denote as NPrecision. NPrecision is meant to render the precision more robust and be comparable across targets of different sizes.

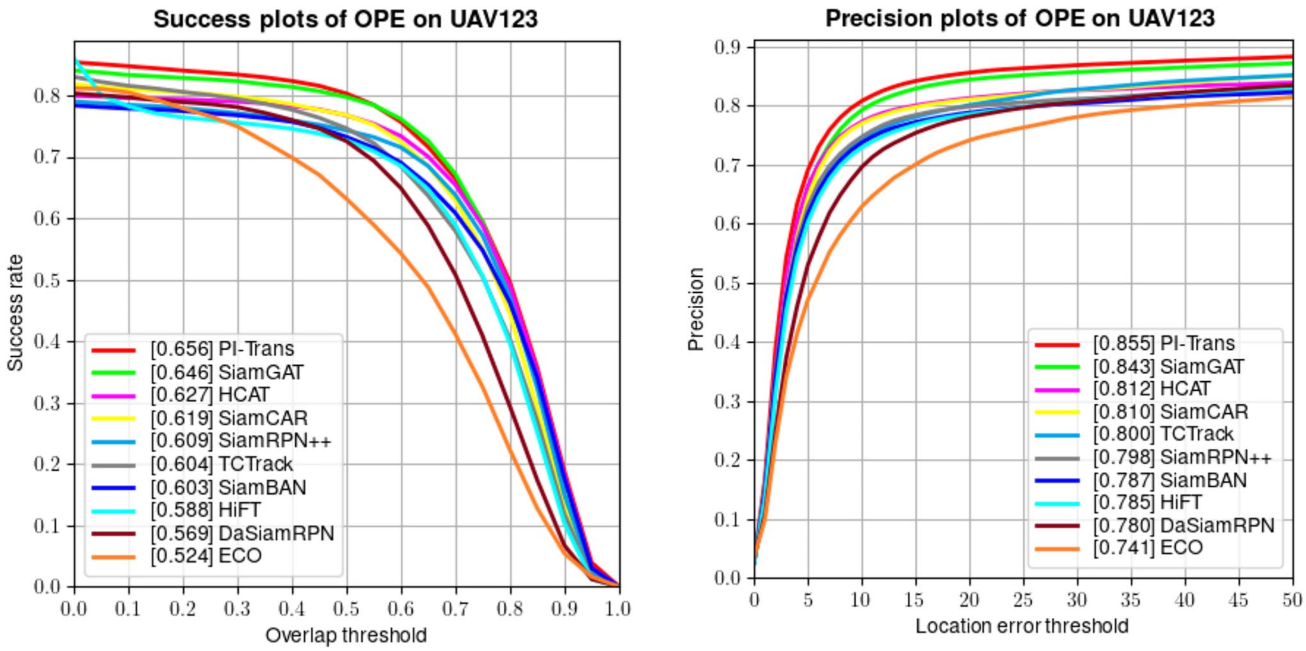

Fig. 6 Performance of PI-Trans and existing SOTA trackers on UAV123 in terms of Success and Precision. It is best viewed when zoomed in

Table 2 The success rate of BC, IV and POC on UAV123

Trackers	*BC*	*IV*	*POC*
SiamRPN++	0.428	0.553	0.515
HiFT	0.396	0.497	0.486
TCTrack	0.397	0.518	0.518
HCAT	0.368	0.580	0.530
PI-Trans	**0.452**	**0.585**	**0.577**

Table 4 shows the performance comparison of PI-Trans against MDNet [10], ECO [9], DiMP [41], SiamFC [13], SiamRPN++ [17], SiamCAR [21], SiamGAT [22], TransT [29] and TT-DiMP [33] across Success, NPrecision and Precision metrics. The results show that PI-Trans exhibits the best performance. In terms of Success, it surpasses the second-ranked TT-DiMP by 0.3% and the third-ranked TransT by 0.5%. The results indicate that PI-Trans has a robust discriminative ability against the distractors and exhibits outstanding generalization capabilities in long-term tracking. The best results are highlighted in bold in Table 4.

4.6 Qualitative analysis and visualization

Figure 7 presents a visual comparison of tracking results on Skating2-1, Liquor and Shaking, and contrast them with the transformer-based trackers HCAT [30] and TransT [29], as well as the Siamese-based tracker SiamFC++ [20]. Skating2-1 encompasses the OCC (occlusion) and DEF (deformation) attributes, Liquor includes the OCC and BC (background clutter) attributes while Shaking features the IV (illumination variation) and BC attributes. It is evident that PI-Trans performs best in these challenging tracking scenarios, with its tracking results more closely aligning

Table 3 Comparison against SOTA trackers on GOT10k

Trackers	*AO*	$SR_{0.5}$	$SR_{0.75}$
Stple	0.246	0.239	0.089
MDNet	0.299	0.303	0.099
ECO	0.316	0.309	0.111
SiamFC	0.374	0.404	0.144
SiamRPN	0.483	0.581	0.270
MTFM	0.506	0.604	0.328
SiamRPN++	0.517	0.416	0.396
IoUformer	0.572	0.642	–
SiamCAR	0.581	0.683	0.441
SiamFC++	0.595	0.695	0.479
DiMP	0.611	0.717	0.492
SiamGAT	0.627	0.743	0.488
PrDiMP	0.634	0.738	0.543
DTT	0.634	0.743	0.488
TT-DiMP	0.640	0.747	0.539
BANDT	0.645	0.749	0.514
HCAT	0.651	0.765	0.567
PI-Trans	0.662	**0.770**	0.583
TransT	**0.671**	0.768	**0.609**

Table 4 Comparison against SOTA trackers on LaSOText

Trackers	*Success*	*NPrecision*	*Precision*
ECO	0.220	0.252	0.240
SiamFC	0.230	0.311	0.269
MDNet	0.279	0.349	0.318
SiamRPN++	0.340	0.416	0.396
SiamCAR	0.351	0.427	0.405
SiamGAT	0.383	0.455	0.410
DiMP	0.392	0.475	0.451
TransT	0.423	–	–
TT-DiMP	0.425	–	–
PI-Trans	**0.428**	**0.504**	**0.482**

with the GT (ground truth). This further demonstrates the stability, robustness and effectiveness of PI-Trans.

Figure 8 shows the tracking performance and tracking speed of PI-Trans, two transformer-based trackers TransT [29] and TCTrack [28], two siamese-based trackers SiamRPN [15] and SiamFC [13], and a correlation filter-based tracker Staple [8] on OTB100. It is evident that although the tracking speed of PI-Trans is lower than TCTrack and SiamRPN, it still meets the requirements for real-time tracking and with better performance. Compared to TransT, PI-Trans is in a leading position.

Although PI-Trans exhibits robust performance, some extreme scenes will cause it to fail in tracking the target. In Fig. 9, the first row shows the target completely disappearing from the field of view, and the second row shows the target being fully occluded. Before encountering these interferences, PI-Trans successfully tracks the target, as seen in #122 of the first row and #102 of the second row. Subsequently, it loses the target, leading to tracking failure. These failure cases indicate that PI-Trans struggles to adapt to rapid changes in the target within a short period, suggesting that there is thus room for improvement in its discriminative and generalization capabilities.

4.7 Ablation studies

In this section, we first conduct ablation studies separately on the encoder, decoder and the position encoding to analyze the effectiveness of each component and then compare the tracking performance when applying different backbone networks. Please note that when conducting an ablation study on one component, the other components remain unchanged.

In Tables 5, 6 and 7, a cross denotes the absence of that component in PI-Trans and we highlight the best values in bold. We use the primary metric AO from GOT10k and the major metric success from OTB100 to illustrate the impact

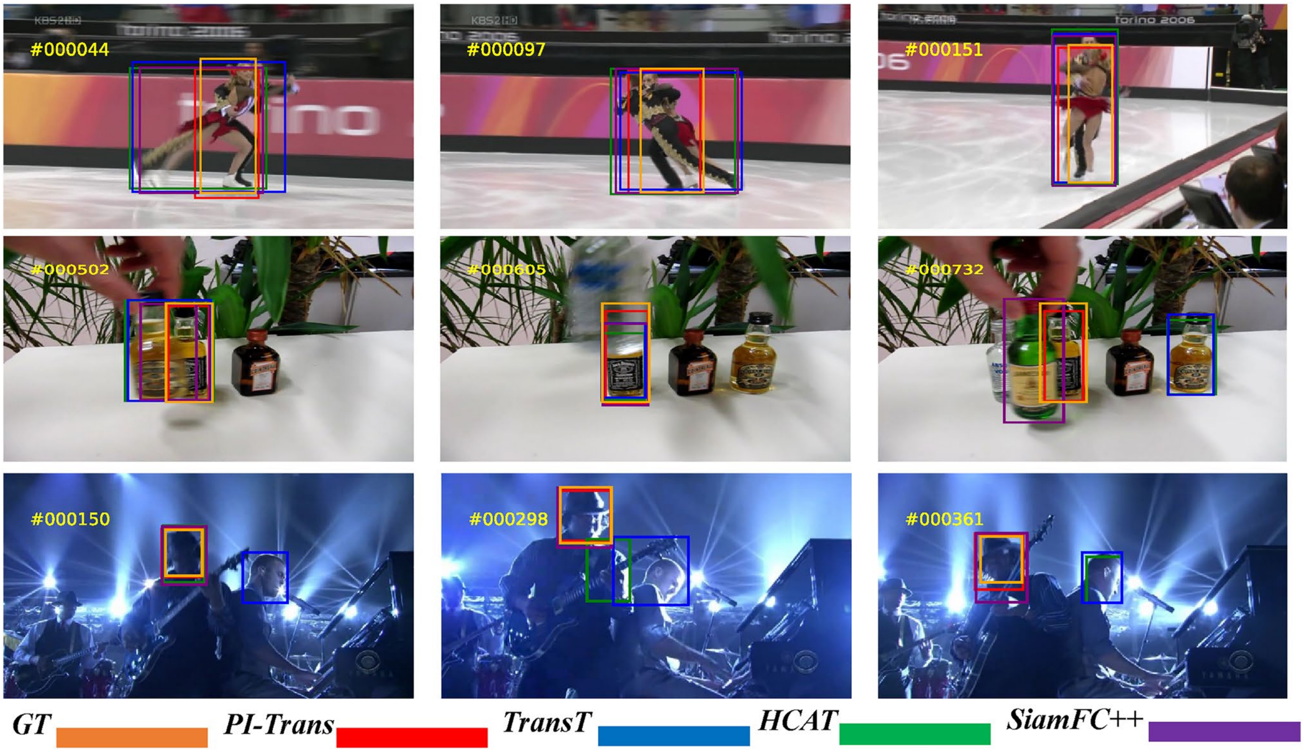

Fig. 7 Visual comparisons of tracking results. The video sequences (from top to bottom) are Skating2-1, Liquor and Shaking. Zoom in for a best view

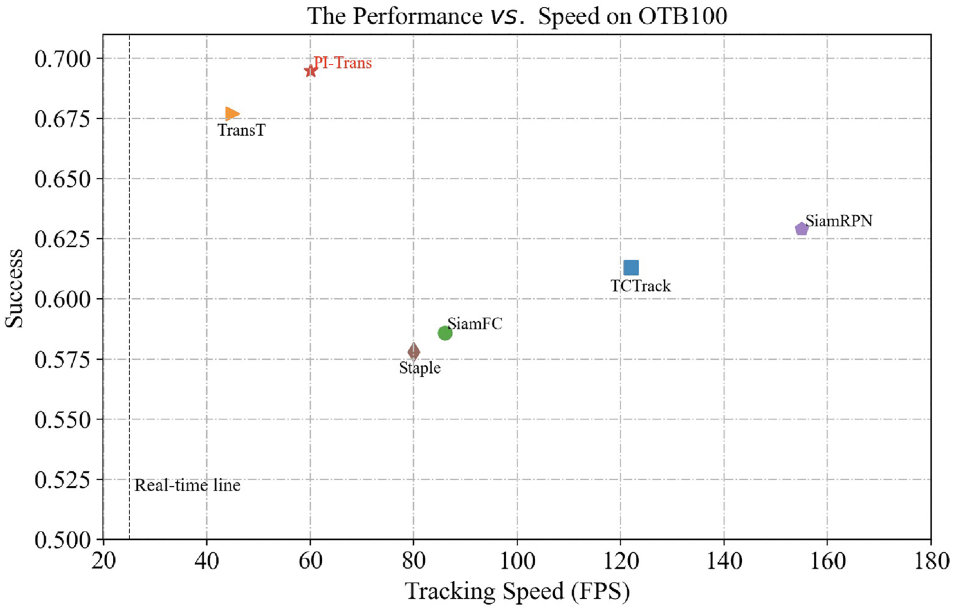

Fig. 8 Tracking performance and tracking speed of the various trackers on OTB100. Zoom in for a best view

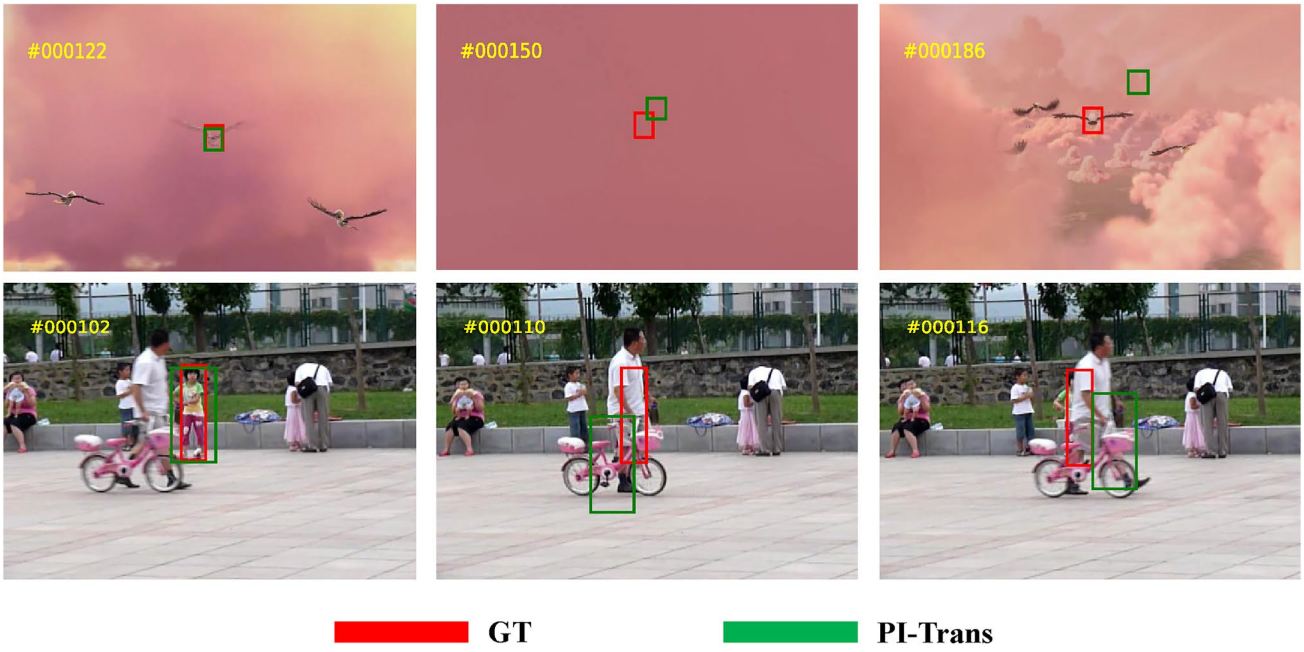

Fig. 9 Failure cases of PI-Trans. GT denotes the ground-truth

of each component of PI-trans on its performance. As can be seen from Table 5, removing the encoder leads to a 5.6% decrease in AO and 5% decrease in success. This means that the introduction of the encoder enables the model to fully explore the global contextual information, acquiring features with strong discriminative ability. This also demonstrates that the interaction between the deep template features and deep search features plays a significant role. When the decoder is removed, AO decreases by 2.5%, while success decreases by 3.2%, indicating that without the integration of the target related prior information, the model struggles to adapt to the different scenarios to accurately distinguish the target from distractions. Finally, the absence of position encoding leads to a 1.2% decrease in AO and 1.3% decrease in success, signifying its beneficial impact on tracking performance.

Table 6 further demonstrates the different inputs to the decoder on performance. We divide the decoder's inputs into

Table 5 Performance of the different components of PI-Trans on GOT10k and OTB100

Encoder	*Decoder*	*Position encoding*	*AO*	*Success*
✗			0.606	0.645
	✗		0.637	0.663
		✗	0.650	0.682
			0.662	**0.695**

Table 6 The impact of different inputs to the decoder on performance across the GOT10k and OTB100

Template features	*Search features*	*AO*	*Success*
✗		0.647	0.683
	✗	**0.662**	**0.695**

Table 7 Performance of different backbone network in PI-Trans on GOT10k and OTB100

Resnet50	*AlexNet*	*AO*	*Success*
✗		0.631	0.658
	✗	**0.662**	**0.695**

template features and search features. It is worth noting that the search image is constantly changing, while the template image remains unchanged. It can be observed that using the latter as input results in a 1.5 % decrease in AO on GOT10k and a 1.2% decrease in success on OTB100 compared to using the former as input. This clearly shows that the prior information obtained from the template features is reliable.

Table 7 shows the impact of different backbone networks on the performance of PI-Trans. It is observed that when using the shallow CNN network AlexNet [14], AO and Success decrease by 3.1% and 3.7%, respectively. This indicates that the target features extracted by the deeper and wider CNN network ResNet50 [43] are more advantageous for our model, enabling it to achieve the best results.

5 Discussion

Table 8 compares PI-Trans and TransT [29] in terms of FLOPs, parameters and the tracking speed. It can be observed that PI-Trans has fewer FLOPs and parameters, making its tracking speed 15 fps faster than TransT. This is because TransT has an additional CFA module designed to fuse features, which increases the model's complexity. Unlike TransT and TrTr [36], PI-Trans's encoder input is the cascade of template and search features, allowing for deep feature interaction. Its decoder input is template features, and the subsequent processing yields prior information that helps the tracker effectively locate the target. In contrast, the decoders of the two mentioned transformer-based trackers take search features as input, and the constantly changing search images increase the interference, potentially leading to target loss. Further reference can be made to Table 6 for the impact of different decoder inputs on tracking performance.

Table 8 Comparison of computational cost and the tracking speed

Trackers	*FLOPs*	*Parameters*	*FPS*
TransT	25.6G	23.0M	45
PI-Trans	16.8G	21.2M	60

From the failure cases in Fig. 9, it can be seen that PI-Trans has certain limitations. It cannot adapt to rapid changes in the target within a short period. Additionally, our feature extraction network still uses CNN. In the future, using the transformer directly to extract more discriminative features could further improve tracking performance. Of course, the novel method of utilizing prior information in PI-Trans can be applied by future researchers in their networks to enhance the network's generalization capability.

6 Conclusion

In this paper, we propose an efficient and robust tracker named PI-Trans, which combines a siamese network with the transformer. The siamese network is used to extract template and search features, which are then cascaded and fed into the transformer's encoder to fully explore the global contextual information. To enable the tracker's adaptation to various tracking scenarios and accurately distinguish the targets from the distractions, we incorporate target-related prior information into the transformer's decoder. Ablation studies further validate the effectiveness of our designed components. PI-Trans achieves a tracking speed of 60 fps, meeting the requirements of real-time applications, and its performance surpasses most SOTA on four major public tracking datasets, i.e, OTB100, UAV123, GOT10k and LaSOText.

Acknowledgements This paper is supported by Key Projects of Heilongjiang Provincial Natural Science Foundation (No. ZD2022F001).

Author contributions Yue Wu: Drafting of the manuscript, software development, methodological planning and verification. Chengtao Cai: Conceptualization, review and supervision. Chai Kiat Yeo: Software development, supervision, editing and review.

Data availability The data are available on reasonable request.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Gao, J., Xu, C.: Learning video moment retrieval without a single annotated video. IEEE Trans. Circuits Syst. Video Technol. **32**(3), 1646–1657 (2021)
2. Gao, J., Chen, M., Xu, C.: Vectorized evidential learning for weakly-supervised temporal action localization. IEEE Trans. Pattern Anal. Mach. Intell. **45**(12), 15949–15963 (2023)
3. Yang, K., Zhao, L., Wang, C.: Workpiece tracking based on improved SiamFC++ and virtual dataset. Multimed. Syst. **29**(6), 3639–3653 (2023)
4. Xue, Y., Jin, G., Shen, T., Tan, L., Wang, N., Gao, J., Wang, L.: Consistent representation mining for multi-drone single object tracking. IEEE Trans. Circuits Syst. Video Technol. 1 (2024)
5. Xue, Y., Jin, G., Shen, T., Tan, L., Wang, N., Gao, J., Wang, L.: Smalltrack: wavelet pooling and graph enhanced classification for UAV small object tracking. IEEE Trans. Geosci. Remote Sens. **61**, 1–15 (2023)
6. Gao, J., Zhang, T., Xu, C.: Learning to model relationships for zero-shot video classification. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3476–3491 (2020)
7. Hu, Y., Gao, J., Dong, J., Fan, B., Liu, H.: Exploring rich semantics for open-set action recognition. IEEE Trans. Multimed. **26**, 5410–5421 (2023)
8. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1401–1409 (2016)
9. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: Eco: efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6638–6646 (2017)
10. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4293–4302 (2016)
11. Xue, Y., Jin, G., Shen, T., Tan, L., Wang, L.: Template-guided frequency attention and adaptive cross-entropy loss for UAV visual tracking. Chin. J. Aeronaut. **36**(9), 299–312 (2023)
12. Xue, Y., Jin, G., Shen, T., Tan, L., Yang, J., Hou, X.: Mobiletrack: Siamese efficient mobile network for high-speed UAV tracking. IET Image Proc. **16**(12), 3300–3313 (2022)
13. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional Siamese networks for object tracking. In: European Conference on Computer Vision, pp. 850–865. Springer (2016)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
15. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with Siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
16. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware Siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 101–117 (2018)
17. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4282–4291 (2019)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. **28**, 91–99 (2015)
19. Chen, Z., Zhong, B., Li, G., Zhang, S., Ji, R.: Siamese box adaptive network for visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6668–6677 (2020)
20. 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, vol. 34, pp. 12549–12556 (2020)
21. Cui, Y., Guo, D., Shao, Y., Wang, Z., Shen, C., Zhang, L., Chen, S.: Joint classification and regression for visual tracking with fully convolutional Siamese networks. Int. J. Comput. Vis. **130**(2), 550–566 (2022)
22. Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., Shen, C.: Graph attention tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9543–9552 (2021)
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems **30** (2017)
24. Xiao, F., Zhang, Z., Yao, Y.: Ctnet: hybrid architecture based on CNN and transformer for image inpainting detection. Multimed. Syst. **29**(6), 3819–3832 (2023)
25. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229. Springer (2020)
26. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
27. Cao, Z., Fu, C., Ye, J., Li, B., Li, Y.: Hift: hierarchical feature transformer for aerial tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15457–15466 (2021)
28. Cao, Z., Huang, Z., Pan, L., Zhang, S., Liu, Z., Fu, C.: Tctrack: temporal contexts for aerial tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14798–14808 (2022)
29. 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, pp. 8126–8135 (2021)
30. Chen, X., Kang, B., Wang, D., Li, D., Lu, H.: Efficient visual tracking via hierarchical cross-attention transformer. In: European Conference on Computer Vision, pp. 461–477. Springer (2022)
31. Yang, K., Zhang, H., Shi, J., Ma, J.: Bandt: a border-aware network with deformable transformers for visual tracking. IEEE Trans. Consumer Electron. **69**, 377–390 (2023)
32. Yu, B., Tang, M., Zheng, L., Zhu, G., Wang, J., Feng, H., Feng, X., Lu, H.: High-performance discriminative tracking with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9856–9865 (2021)
33. Nie, J., Wu, H., He, Z., Gao, M., Dong, Z.: Spreading fine-grained prior knowledge for accurate tracking. IEEE Trans. Circuits Syst. Video Technol. **32**(9), 6186–6199 (2022)
34. Lu, X., Wang, Z., Wang, X., Hei, X.: Multi-template temporal information fusion for Siamese object tracking. IET Comput. Vis. **17**(1), 51–61 (2023)
35. Cai, H., Lan, L., Zhang, J., Zhang, X., Zhan, Y., Luo, Z.: Iouformer: pseudo-IoU prediction with transformer for visual tracking. Neural Netw. **170**, 548–563 (2024)
36. Zhao, M., Okada, K., Inaba, M.: Trtr: visual tracking with transformer. arXiv preprint arXiv:2105.03817 (2021)
37. Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: a benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)

38. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for UAV tracking. In: European Conference on Computer Vision, pp. 445–461. Springer (2016)
39. 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. **43**(5), 1562–1577 (2021)
40. Fan, H., Bai, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Huang, M., Liu, J., Xu, Y., et al.: Lasot: a high-quality large-scale single object tracking benchmark. Int. J. Comput. Vis. **129**(2), 439–461 (2021)
41. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6182–6191 (2019)
42. Danelljan, M., Gool, L.V., Timofte, R.: Probabilistic regression for visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7183–7192 (2020)
43. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
44. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
45. Huang, Z., Zhang, S., Pan, L., Qing, Z., Tang, M., Liu, Z., Ang Jr, M.H.: Tada! Temporally-adaptive convolutions for video understanding. In: ICLR (2022)
46. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, pp. 315–323 (2011)
47. 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, pp. 2980–2988 (2017)
48. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-IoU loss: faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12993–13000 (2020)
49. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: European Conference on Computer Vision, pp. 740–755. Springer (2014)
50. 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. **115**(3), 211–252 (2015)
51. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: a high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5374–5383 (2019)
52. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.