International Journal of Network Dynamics and Intelligence Scilight

Article

Learning Regression Distribution: Information Diffusion from Template to Search for Visual Object Tracking

Shuo Hu*, Jinbo Lu, and Sien Zhou

School of Electrical Engineering, Yanshan University, Qinhuangdao 066000, China
* Correspondence: hus@ysu.edu.cn

Received: 5 July 2023
Accepted: 19 October 2023
Published: 26 March 2024

Abstract: The general paradigm of traditional Siamese networks involves using cross-correlations to fuse features from the backbone, and this paradigm is limited by the inductive bias of the convolution kernel, resulting in the lack of global information. In this paper, we propose the Siamese learning regression distribution (SiamLRD) to address the local limitations of traditional cross-correlation operations on feature fusion and weak self-connections between features within different branches. The SiamLRD uses the cross-attention mechanism to replace cross-correlations between the features of the target region of interest and the template so as to enhance flexibility. Firstly, the original transformer structure is improved to be suitable for convolutional Siamese networks. The improved transformer architecture is then used to replace cross-correlation operations, resulting in more comprehensive feature fusion between branches. Secondly, we introduce a new decoder structure into the novel fusion strategy to enhance the correlation between classification scores and regression accuracy during decoding. Multiple benchmarks are used to test the proposed SiamLRD approach, and it is verified that the proposed approach improves the baseline with 5.8% in terms of AO and 9.7% in terms of $\mathrm{SR}_{0.75}$ on the GOT-10K dataset.

Keywords: Siamese network; fusion strategy; learning regression distribution; transformer

1. Introduction

Visual object tracking aims to estimate the state of a specified object in a video sequence, which is a fundamentally challenging task in computer vision. With the emergence of deep learning-based tracking algorithms [1−7], the performance of trackers has been greatly improved. Although those new algorithms have broken through the bottleneck of traditional algorithms, their performance potential has not yet been fully explored.

Single-object tracking is a crucial important branch in the field of computer vision. Siamese feature fusion (SiamFC) [1] formulates the tracking task as a matching problem based on semantic features, but its feature fusion method is incomplete due to direct cross-correlations between feature maps of two branches. By introducing the region proposal network (RPN) strategy, SiamRPN [2] improves the feature fusion mechanism. SiamRPN++ [3], proposed in 2018, innovatively solves the destruction problem of the strict translation invariance by using an effective spatial awareness sampling strategy. SiamCAR [5], proposed in the same year, aims to reduce the number of parameters. In SiamCAR, the anchor-free idea is introduced and a center-ness auxiliary branch is added to the decoder heads, which alleviates the low correlation between the regression branch and the classification branch in other trackers.

Convolutional neural networks (CNNs) have the inherently strong local inductive bias characteristic of the convolution kernel. This characteristic speeds up the convergence of the models, but limits the performance of the convolutional network models. One reason is that the features extracted by the convolutional models lack global information [8]. Recently, attention structures have been introduced into the field of deep learning. DETR [9] initially introduces the global information processing module into the field of computer vision, but its convergence is slow due to the fact that the decoder treats randomly initialized variables as queries. ViT [10] directly carries out global modeling in the process of feature extraction, making the model-based transformer [11] structure to have comparable performance compared with the convolutional model. Note that the ViT needs more data for training and contains a large number of parameters, resulting in higher computation costs and storage requirements. Therefore, we retain the fully convolutional feature extraction network and focus on optimizing the feature fusion stage to improve model performance.

Our model uses CNN as the feature extraction network and adopts the transformer as the feature fusion module. Subsequently, it is hard to conduct optimization because these two structures employ different learning rate warm-up strategies. Inspired by [12], we introduce the Pre-LN into our structure to alleviate this conflict and unify the training process.

The motivation of the proposed SiamLRD is to combine global information in order to learn more accurate general boundary distributions. In conclusion, our contributions are summarised as follows.

- An encoder-decoder structure is proposed to fuse the dual-branch backbone feature maps of the Siamese network to replace the original depthwise cross-correlation operation.
- Regarding the modified feature fusion strategy, for the regression branch, we guide the model to learn a general distribution of object boundaries with acceptable computing costs.
- In order to ensure consistency between the decoding processes of training and testing, we remove auxiliary detection heads and enhance the consistency between training and testing by expanding the presentation of the classification branch's output.
- To ensure the reliability of the training process, a Pre-LN strategy is introduced to alleviate the conflict between the two structures of the training process.

2. Related Work

2.1. The Siamese Tracker

The Siamese network structure was first introduced to solve single-object tracking problems in SiamFC [1]. To enable the model to recognize the tracked object, a set of video frames, consisting of a template and search frames, is input into the Siamese network model based on the weight-sharing strategy. In essence, the Siamese network has only one backbone network [1]. In the tracking process, the backbone network is utilized to process both the template frame and the current frame separately, and this process is figuratively called weight sharing. A noteworthy point is that SiamFC's feature fusion method is incomplete due to direct cross-correlations between feature maps of two branches. To further improve performance, SiamRPN [2] was proposed to improve the feature fusion strategy of SiamFC by replacing the direct feature fusion mechanism of the two branches with an RPN mechanism. Nevertheless, the backbone of SiamRPN is shallow, and its feature extraction capability is limited. Therefore, SiamRPN++ [3] was proposed to successfully integrate ResNet [13] into the Siamese network framework by introducing a spatial awareness strategy to solve the limitation problem of network depth.

Based on the anchor-free approach, SiamCAR [5] was proposed in 2018 to reduce the number of model parameters after introducing the deeper backbone and avoid the complex hyperparameter optimization problems associated with anchor boxes. SiamCAR uses multi-stage depthwise cross-correlations to replace the RPN mechanism, thereby improving the tracking speed, maintaining the accuracy and simplifying the model structure. In this paper, we consider SiamCAR as our baseline model.

2.2. Vision Transformer

Recently, the attention structure was introduced into the field of tracking. A global information model was proposed in SiamGAT [6] with graph attention structures such as DiMP [14], TrDiMP [15], and SiamAttn [7], where attention operations and even transformer modules were directly introduced to replace cross-correlation operations in the Siamese network architecture. The transformer [11] was originally proposed in the field of natural language processing to solve the problem of RNN parallelization as well as explicitly build global modeling capabilities. The transformer can adapt to input changes because its attention operation is dynamic. ViT [10] was proposed to serialize images through convolution, where the transformer was introduced as a feature extraction network for image processing. Note that the ViT models the image in a global receptive field and extracts the global information of the image.

3. Method

The structure of the proposed model is illustrated in Figure 1 which primarily comprises three parts: a feature extraction network, a feature fusion network, and a decoding detection head. Among them, we mainly optimize the feature fusion network and the decoding detection head, and the optimization process will be discussed in Subsections 3.2 and 3.3.

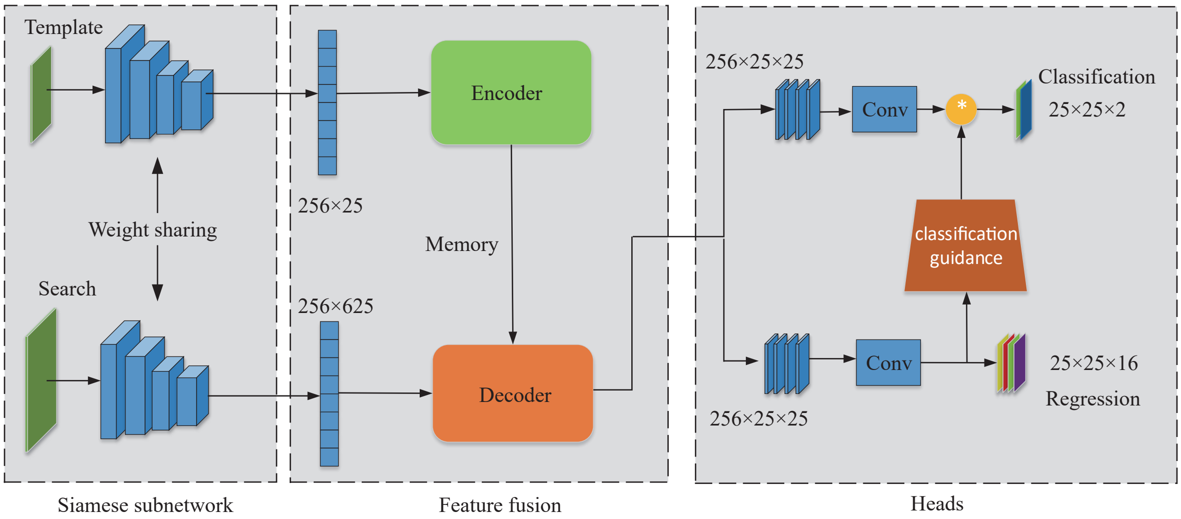

Figure 1. The architechture of the proposed model. The feature fusion uses the transformer to replace depthwise cross-correlations. Heads indicate the general distribution prediction module that is compatible with feature fusion.

3.1. Feature Extraction Network

In this paper, the ResNet50 is replaced by the lighter GoogLeNet [16] to act as the backbone network. Additionally, we remove the pooling layer from the second stage and the fifth stage of the GoogLeNet to ensure that the size of the feature map is the same as that of the baseline model. In addition, since the spatial structure of the model needs to be retained and the penalty coefficient is used in the detection head, the outermost four pixels of the feature map are clipped without loss of performance, as shown in Equation (1). The crop operation also helps to reduce the interference of background clutters.

$$crop(x) = x[:,:,4:h-5,4:w-5] \quad x \in R^{b\times c\times h\times w}, \tag{1}$$

where h and w are the spatial sizes of the inputs, and b and c represent, respectively, the batch size and the channel of the inputs. Since the maximum indices of feature map pixels in the spatial dimension are $h-1$ and $w-1$, the maximum indices for clipping are $h-5$ and $w-5$.

3.2. Attention

The transformer gains the flexible modeling ability under the global context due to the attention mechanism. Self-attention projects the input into three domains, namely Q, K and V, through three projection matrices. The attention weight is calculated by multiplying the matrices Q and K, and then applying a softmax function to normalize the attention. After calculating the attention weight map, matrix multiplication is performed between the weight and V to obtain the output of the attention module. This mechanism enables the model to have the capability of global context modeling. The calculation process is as follows:

$$\begin{gathered} Q = xW^q, K = xW^k, V = xW^v; \\ A^{(m)} = softmax\left(\frac{Q^{T(m)}K^{(m)}}{\sqrt{d}}\right); \qquad (2) \\ Attn(Q,K,V) = Concat(A^{(1)}V^{(1)},\cdots,A^{(n)}V^{(n)})W^o, \end{gathered}$$

where $W^q \in R^{C_{in}\times C_{dim}}$, $W^k \in R^{C_{in}\times C_{dim}}$, and $W^v \in R^{C_{in}\times C_{dim}}$ are linear projection layers. Q, K and V, respectively, represent the query, key and value. Meanwhile, n means the total number of attention heads, $A^{(m)}$ is the mth attention map and d is the dimension of x after projection. $W^o \in R^{C_{dim}\times C_{out}}$ is used for further processing of multi-head attention after concatenation.

Unlike other visual algorithms based on the transformer architecture, the method proposed in this chapter does not use the vanilla transformer. The reason is that algorithms based on the vanilla transformer are usually optimized by the Adam [17] optimizer or the AdamW [18] optimizer. Note that the backbone usually needs to be completely unfrozen in the early stage, which is incompatible with the training process of the CNN backbone network. Meanwhile, due to the slow convergence speed of the vanilla transformer, more epochs are performed for training which significantly slows down the training speed of the model. Therefore, in this chapter, we adopt the improved transformer structure. Specifically, we introduce the LN layer before the attention operation in order to eliminate the learning rate warm-up of the transformer and harmonize the training strategies of the CNN and the transformer architecture. The improved structure is called as the Pre-LN transformer, as shown in Figure 2.

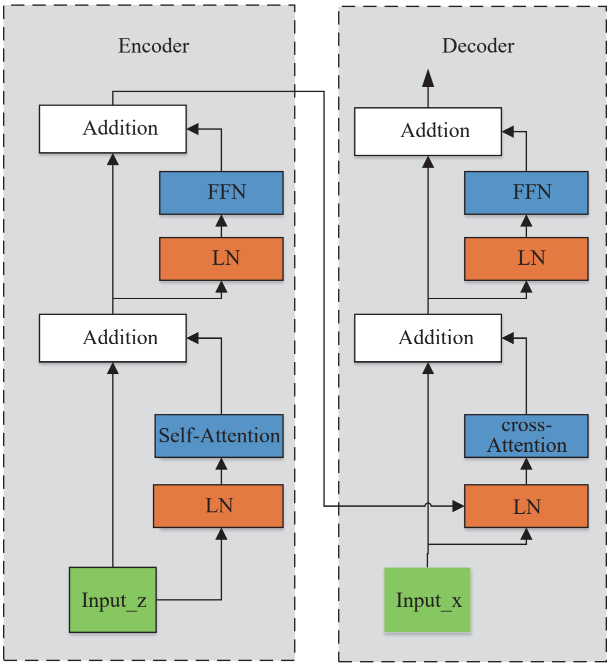

Figure 2. The structure of the Pre-LN transformer.

The model proposed in this paper inserts the transformer module into two branches of the Siamese network. Specifically, the encoder in the template branch and the decoder in the search branch. The specific structure is shown in Figure 1. For the encoder, we take the flattened output of the template branch and the absolute positional encoding as the input of the encoder, as shown in Equation (3). For the decoder placed in the search branch, we modify the inputs. One of the inputs, which comes from the encoder, is used to produce K an V. For the other input, we use the flattened sequence of the feature map output from the search branch rather than directly choosing an initialized learnable sequence. We first perform a self-attention operation in the feature map of the search branch output. Then, we perform a cross-attention operation between the output of the previous step and the encoder's output.

$$Q^p = Q + pos, K^p = K + pos, V^p = V, \tag{3}$$

where pos represents positional encoding, and Q^p and K^p mean that they have been added to the positional encoding. Generally, V does not add positional encoding, so $V = V^p$.

In addition, our model combines both CNN and transformer structures. Considering the computing costs, we simplify the transformer structure to a single layer of the encoder and decoder in the feature fusion stage. The CNN remains the primary component of the model, while the transformer serves as a feature fusion component. Therefore, the main hyperparameters of the model are from the CNN architecture. To address the impact of hyperparameter divergence caused by learning rate warm-up, the position of the layer normalization needs to be adjusted.

$$T(x) = LN(FFN(LN(x + attn(x)))), \tag{4}$$

where LN stands for the layer normalization, and FFN represents the fully connected feed-forward network.

In Pre-LN [12], the experimental results show that the position of layer normalization has a significant impact on the learning rate strategy during the training stage. Specifically, pre-positioning the layer normalization can prevent the warm-up of the learning rate of the transformer. In this paper, Pre-LN is used to replace the original Post-LN, as shown in equation (4), in order to eliminate the requirement for the learning rate warm-up hyperparameter of the transformer. The hyperparameter divergence is prevented. The modified process is represented by the following Equation:

$$T(x) = FFN(LN(x + attn(LN(x)))). \tag{5}$$

3.3. Head Subnetwork

For the regression branch, the feature fusion subnetwork uses a fusion method based on the attention mechanism to enable the model to comprehend the global image information during training. Additionally, the existence of self-attention helps the model gain a more comprehensive understanding of the distribution of training data. This motivates us to use a distribution-friendly detection head to regress the boundary of the tracked object. Inspired by [19], we abandon the optimization of the Dirac distribution, because information presented by the Dirac distribution has been compressed. Instead, we optimize a more comprehensive distribution that fully utilizes the advantages of the attention mechanism. Furthermore, an issue of blurred boundaries exists in the dataset. Therefore, optimizing a general distribution instead of a Dirac distribution can alleviate the impact of blurred boundaries in the dataset.

The predicted bounding box can be represented by the distances from the corresponding location to the four sides of the bounding box in the input search region. We no longer optimize the general distribution based on the four distances of the left (l), top (t), right (r), and bottom (b), as in the case of the Dirac distribution. Instead, we directly optimize the discrete probabilities of the distances on the left (P_l), top (P_t), right (P_r), and bottom (P_b). For the improved model, the bounding box can be calculated from the corresponding discrete probabilities as follows:

$$y = \sum_{i=0}^{n} P_y(x_i)x_i \tag{6}$$

where $y \in [l,t,r,b]$, $P_y(x)$ represents the probability that the location is a boundary when the distance is x_i, and x_i represents the distance from the current pixel. We apply the distribution focal loss (DFL) [19] to guide the discrete probability distributions. Additionally, we employ the GIOU loss [20] to optimize the bounding box obtained from Equation (6).

$$L_p = \sum_{h,w} -\frac{1}{4} \sum_{i \in [l,t,r,b]} ((x_{i,2} - target_i)\log(P_{i,2}) + (target_i - x_{i,1})\log(P_{i,1})) \tag{7}$$

where $x_{i,*}$ represents the distance from the current position to the boundary box, $x_{i,1}$ and $x_{i,2}$ represent the two discretized distances which are closest to the target boundary, $P_{i,*}$ represents the probability of $x_{i,*}$, and $target_i$ is the boundary label.

For the classification branch, we remove the auxiliary detection head, and extend the meaning of the classifier's labels to enhance the correlation between the learning processes of the two branches.

Specifically, we expand the meaning of the classification branch directly in order to address the differences in processing classification response maps during the training and testing stages of the auxiliary branch. This not only enables the classification branch to learn the discriminative function between the tracked object and background, but also allows direct prediction of the regression branch and regression quality. The proposed method enhances the correlation between the knowledge of the two branches, guiding the classification branch to focus on the intersection over union (IOU) between the regression bounding box and the ground truth.

The calculation process of IOU is as follows:

$$IOU = \frac{A \cap B}{A \cup B} \tag{8}$$

where A and B represent the predicted bounding box and ground truth bounding box, respectively.

Drawing inspiration from IOUNet [21], we first extend the discrete logical variables (that represent the target foreground and background) to continuous value domains. As shown in Equation (9), for the background classification label, we uniformly define it as 0. Our work mainly expands the classification labels of the foreground information of the tracked target. Specifically, the foreground labels of the classification branch are the IOU values between the regression bounding box and the ground truth. The values of classification labels describe the confidence coefficient by which the target is classified as the foreground and the quality of the predicted results of the regression branch at the current position.

$$y_{label} = \begin{cases} y_{cls} * IOU, & \text{if } y_{cls} \neq 0 \\ 0, & \text{if } y_{cls} = 0 \end{cases} \tag{9}$$

where y_{label} represents the continuous classification label, and y_{cls} represents the discrete classification label.

For the classification branch after continuous processing, BCELoss cannot be applied for training. In this paper, we introduce the quality focal loss [19] to address the training problem, as shown in Equation (10).

$$QFL = -|y_{label} - \tilde{y}|^{\beta}((1 - y_{label})\log(1 - \tilde{y}) + y_{label}\log(\tilde{y})) \tag{10}$$

where y_{label} is the continuous classification label, $\tilde{y}$ is the prediction of the classification branch, and β is a tunable parameter used to control the down-weighting rate smoothly.

To further enhance the correlation between the classification branch and the learned boundary distribution, we introduce a classification guidance module based on the method proposed in [22], as shown in Figure 3. This module incorporates the learned distribution from the regression branch into the classification branch to influence the classification results. The structure of the classification guidance module is composed of two convolutional layers, with the ReLU activation function applied after the first layer. Specifically, we use the TOP-K method (where K is set to be 4) to obtain boundary distribution information from the regression branch. This information is processed by the classification guidance module and then multiplied by the classification logits, yielding the final classification score map. The specific process is shown in Equation (11).

$$\begin{gathered} CG = Conv(Relu(Conv(topk(f)))); \\ CScore = Cls * CG \end{gathered} \tag{11}$$

where f represents the output of the regression branch, CG represents the output of the classification guidance module, and $CScore$ represents the enhanced classification score map.

The improved loss function is as follows:

$$Loss_{title} = l_{QFL} + \lambda_1 * L_1 + \lambda_2 * l_{giou} \tag{12}$$

where λ_1 and λ_2 are the weights of the loss function, which are set to be 0.25 and 1 by default. $Loss_{title}$ is the sum of the two branch loss functions, l_{QFL} is the classification loss, L_1 is the L1 loss, and l_{giou} is the GIOU loss.

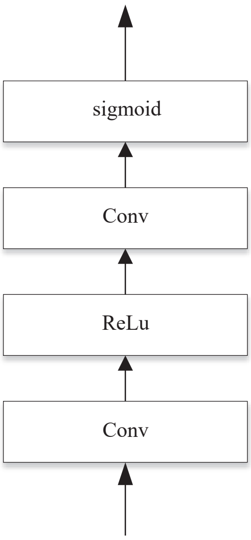

Figure 3. Classification guidance module.

4. Experiments

4.1. Experimental Details

Each version of the model proposed in this paper is deployed on a server installed with two GPUs (1080ti). The number of training epochs is consistent with the baseline model. The training strategy is as follows: the base learning rate is set to be 0.1, the initial learning rate is set to be 0.02, and the learning rate is warm-up to be 0.1 during the first five epochs. The cosine annealing strategy is used to adjust the learning rate in the next 15 epochs. The learning rate of the backbone network is set to be 10% of that of the other modules. The optimizer is SGD, and the weight decay and momentum settings are consistent with the baseline model. The parameters of the backbone are unfrozen at the 11th epoch. In addition, the training datasets are the same as those in SiamCAR [5]. The models are evaluated on three public benchmarks: GOT-10K [23], LaSOT [24] and LaSOText [24].

4.2. Evaluation Results on GOT-10K

GOT-10K is a widely-used benchmark for evaluating the tracking performance of object tracking models. It contains over 10000 video sequences. The evaluation process is conducted online to ensure the fairness of the testing process, and the labels of the test set are not publicly available. In the evaluation protocol, the training process can only use the train set split from GOT-10K. We comply with this requirement and compare our model with the state-of-the-art trackers. As shown in Figure 4 and Table 1, our model surpasses SiamCAR [5] by 5.8% in terms of AO, by 5.8% in terms of $\mathrm{SR}_{0.5}$ and by 9.7% in terms of $\mathrm{SR}_{0.75}$, outperforming other public trackers on the GOT-10K dataset.

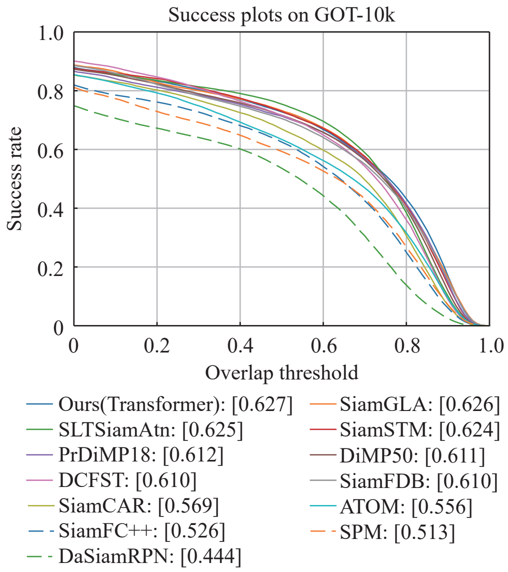

Figure 4. Success plot on GOT-10K [23].

Table 1 The evaluation on GOT10k [23] benchmark

Tracker	AO↑	SR$_{0.5}$↑	SR$_{0.75}$↑
DaSiamRPN [25]	44.4	53.6	22.0
SPM [26]	51.3	59.3	35.9
SiamFC++ [27]	52.6	62.5	34.7
ATOM [28]	55.6	63.4	40.2
SiamCAR [5]	56.9	67.0	41.5
SiamFDB [29]	61.0	70.5	48.2
DCFST [30]	61.0	71.6	46.3
DiMP50 [14]	61.1	71.7	49.2
PrDiMP18 [31]	61.2	71.3	50.3
SiamSTM [32]	62.4	73.0	50.3
SLTSiamAtn [33]	62.5	75.4	50.1
SiamGLA [34]	62.6	73.2	50.5
Ours (Transformer)	62.7	72.8	51.2

Values marked by red, green, and blue represent the order of each indication from the first to the third columns.

4.3. Evaluation Results on LaSOT

LaSOT dataset [24] is a challenging dataset of long-term video sequences designed for object tracking. The dataset contains more than 1400 video sequences and approximately 35000 frames with more than 280 video sequences for testing. Our model is compared with the baseline tracker and the state-of-the-art trackers. As shown in Figure 5 and Table 2, the precision is improved by 2.0% compared with SiamCAR [5], and the success rate is increased by 1.9%. In addition, the success rate of the model proposed in this paper is slightly lower than that of LTMU [35], and the precision is higher than that of LTMU, which is due to the contribution of the classification guidance module. In a word, the proposed tracker outperforms other public trackers.

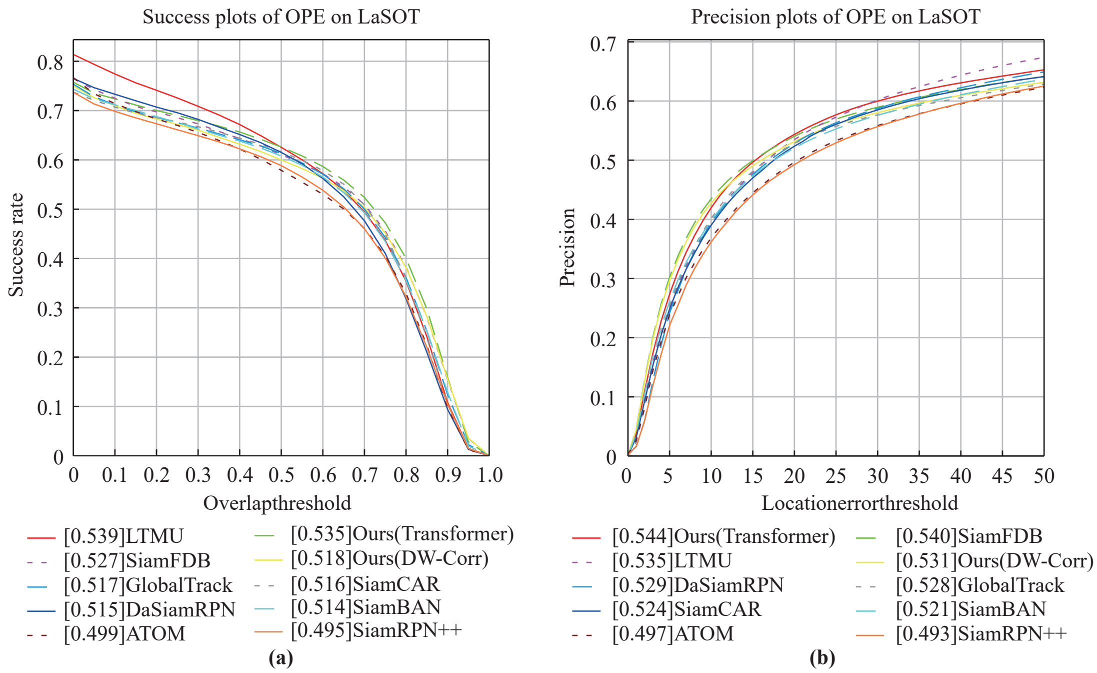

Figure 5. Contrast plot on LaSOT [24]. **(a)** Precision plot. **(b)** Success plot.

Table 2 The evaluation on LaSOT [24] benchmark

Tracker	Precision↑	Success↑
SiamRPN++ [3]	0.493	0.495
ATOM [28]	0.497	0.499
SiamBAN [4]	0.521	0.514
SiamCAR [5]	0.524	0.516
GlobalTrack [36]	0.528	0.517
DaSiamRPN [25]	0.529	0.515
Ours(DW-XCorr)	0.531	0.518
LTMU [35]	0.535	0.539
SiamFDB [29]	0.540	0.527
Ours (Transformer)	0.544	0.535

Values marked by red, green, and blue represent the orders of each indication from the first to the third columns.

4.4. Evaluation Results on LaSOText

LaSOText dataset is an extended benchmark for LaSOT. Different from LaSOT [24], LaSOText uses all 1400 video frame sequences from LaSOT as the training set and the 15 categories that are not related to the original LaSOT (150 video frame sequences in total) as the test set. This enables the LaSOText benchmark to detect model generalization more effectively than LaSOT. As shown in Figure 6 and Table 3, our model achieves a success rate of 35.0% and outperforms other trackers.

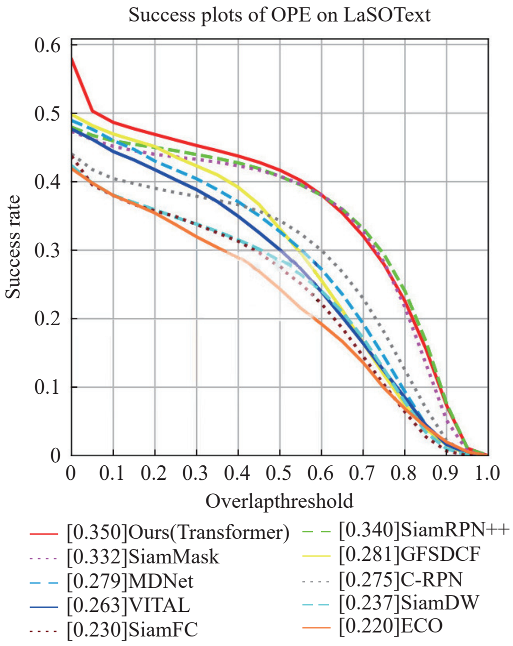

Figure 6. Success plot on LaSOText [24].

Table 3 The evaluation on LaSOText [24] benchmark

Tracker	Success↑
ECO [37]	0.220
SiamFC [1]	0.230
SiamDW [38]	0.237
VITAL [39]	0.263
C-RPN [40]	0.275
MDNet [41]	0.279
GFSDCF [42]	0.281
SiamMask [43]	0.332
SiamRPN++ [3]	0.340
Ours (Transformer)	0.350

Values marked by red, green, and blue represent the orders of each indication from the first to the third columns.

4.5. Ablation Study

To validate the effectiveness of the proposed model, we conduct ablation experiments to compare the impact of different structures. The following groups of experimental data are mainly compared. The transformer feature fusion module in our model is replaced by a double-branch cross-correlation module to create a new model that is compared with our model and the baseline model. SiamCAR-ours represents that the baseline model is trained by our hardware, and SiamCAR-B represents the data presented in [5]. Ours (DW-XCorr) means to change the transformer feature fusion module into a double-branch (corresponding to two detection heads) cross-correlation operation. Ours (transformer) represents the model proposed in this paper.

As shown in Table 4, the proposed model surpasses the two-branch feature fusion model by 4.1% and the baseline model by 5.8% in terms of AO. The $SR_{0.5}$ gains improvement of 5.8% achieving the second place and 7% achieving the third place. In addition, the transformer architecture used in our model is a single layer version, showing that the proposed model achieves better performance with a similar parameter number.

Table 4 The ablation study on GOT10k [23] benchmark

Tracker	AO↑	$SR_{0.5}$↑	$SR_{0.75}$↑
SiamCAR-ours	54.7	63.5	42.6
SiamCAR-B [5]	56.9	67.0	41.5
Ours(DW-XCorr)	58.6	65.8	43.4
Ours (Transformer)	62.7	72.8	51.2

Values marked by red, green, and blue represent the orders of each indication from the first to the third columns.

5. Conclusion

In this paper, we have proposed a network structure that integrates a CNN and a transformer module to combine the strengths of both. The effectiveness of the proposed structure has been fully demonstrated by experiments. These experiments have also indirectly proven that existing CNNs lack global information. Compared to the case where the transformer is used as the backbone network model, the proposed model has the problem of a sharp decline in regression accuracy under complex backgrounds. To alleviate this problem, in the future, we will explore the features extracted by the convolutional backbone network and the transformer backbone network, respectively.

Author Contributions: Shuo Hu: data acquisition, thesis review and guidance, project management, article correction; **Jinbo Lu**: method implementation, method testing, data processing, the member of writing group; **Sien Zhou**: method testing, data processing and labeling, the member of writing group.

Funding: This work is supported by National Natural Science Foundation of China (No.62073279), and the Key Projects of Hebei Natural Science Foundation [F2021203054], and the Central Guidance on Local Science and Technology Development Fund of Hebei Province (216Z2001G), and the Hebei innovation capability improvement plan project [22567619H].

Data Availability Statement: Not applicable.

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

References

1. Bertinetto, L.; Valmadre, J.; Henriques, J. F.; *et al*. Fully-convolutional siamese networks for object tracking. In *Proceedings of European Conference on Computer Vision, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016*; Springer: Berlin/Heidelberg, 2016; pp. 850–865. doi: 10.1007/978-3-319-48881-3_56
2. Li, B.; Yan, J. J.; Wu, W.; *et al*. High performance visual tracking with siamese region proposal network. In *Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018*; IEEE: New York, 2018; pp. 8971–8980. doi: 10.1109/CVPR.2018.00935
3. Li, B.; Wu, W.; Wang, Q.; *et al*. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019*; IEEE: New York, 2019; pp. 4277–4286. doi: 10.1109/CVPR.2019.00441
4. Chen, Z. D.; Zhong, B. N.; Li, G. R.; *et al*. Siamese box adaptive network for visual tracking. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, 2020; pp. 6667–6676. doi: 10.1109/CVPR42600.2020.00670
5. Guo, D. Y.; Wang, J.; Cui, Y.; *et al*. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, 2020; pp. 6268–6276. doi: 10.1109/CVPR42600.2020.00630
6. Guo, D. Y.; Shao, Y. Y.; Cui, Y.; *et al*. Graph attention tracking. In *Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021*; IEEE: New York, 2021; pp. 9538–9547. doi: 10.1109/CVPR46437.2021.00942
7. Yu, Y. C.; Xiong, Y. L.; Huang, W. L.; *et al*. Deformable Siamese attention networks for visual object tracking. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, 2020; pp. 6727–6736. doi: 10.1109/CVPR42600.2020.00676
8. Zhao, M. J.; Okada, K.; Inaba, M. TrTr: Visual tracking with transformer. arXiv: 2105.03817, 2021. doi: 10.48550/arXiv.2105.03817
9. Carion, N.; Massa, F.; Synnaeve, G.; *et al*. End-to-end object detection with transformers. In *Proceedings of the 16th European Conference on Computer Vision*, *Glasgow*, *UK*, *23–28 August 2020*; Springer: Berlin/Heidelberg, 2020; pp. 213–229. doi: 10.1007/978-3-030-58452-8_13
10. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; *et al*. An Image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations, 3–7 May 2021*; OpenReview. net, 2021.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; *et al*. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach*, *CA*, *USA, 04 December 2017*; Curran Associates Inc. : Red Hook, 2017; pp. 6000–6010. doi: 10.5555/3295222.3295349
12. Xiong, R. B.; Yang, Y. C.; He, D.; *et al*. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, *13 July 2020*; PMLR, 2020; p. 975.
13. He, K. M.; Zhang, X. Y.; Ren, S. Q.; *et al*. Deep residual learning for image recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, *Las Vegas*, *NV*, *USA*, *27–30 June 2016*; IEEE: New York, 2016; pp. 770–778. doi: 10.1109/CVPR.2016.90
14. Bhat, G.; Danelljan, M.; van Gool, L.; *et al*. Learning discriminative model prediction for tracking. In *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision*, *Seoul*, *Korea* (*South*), *27 October 2019–02 November 2019*; IEEE: New York, 2019, pp. 6181–6190. doi: 10.1109/ICCV.2019.00628
15. Wang, N.; Zhou, W. G.; Wang, J.; *et al*. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Nashville, TN, USA, 20–25 June 2021*; IEEE: New York, 2021, pp. 1571–1580. doi: 10.1109/CVPR46437.2021.00162
16. Szegedy, C.; Liu, W.; Jia, Y. Q.; *et al*. Going deeper with convolutions. In *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition*, *Boston*, *MA*, *USA*, *07–12 June 2015*; IEEE: New York, 2015; pp. 1–9. doi: 10.1109/CVPR.2015.7298594
17. Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning*

Representations, *San Diego*, *CA*, *USA*, *7–9 May 2015*; ICLR, 2015.
18. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. arXiv: 1711.05101, 2019. doi: 10.48550/arXiv.1711.05101
19. Li, X.; Lv, C. Q.; Wang, W. H.; *et al*. Generalized focal loss: Towards efficient representation learning for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell., **2023**, *45*: 3139−3153. doi: 10.1109/TPAMI.2022.3180392
20. Rezatofighi, H.; Tsoi, N.; Gwak, J. Y.; *et al*. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Long Beach*, *CA*, *USA*, *15–20 June 2019*; IEEE: New York, 2019; pp. 658–666. doi: 10.1109/CVPR.2019.00075
21. Jiang, B. R.; Luo, R. X.; Mao, J. Y.; *et al*. Acquisition of localization confidence for accurate object detection. In *Proceedings of the 15th European Conference on Computer Vision*, *Munich*, *Germany*, *8–14 September 2018*; Springer: Berlin/Heidelberg, 2018; pp. 816–832. doi: 10.1007/978-3-030-01264-9_48
22. Li, X.; Wang, W. H.; Hu, X. L.; *et al*. Generalized focal loss V2: Learning reliable localization quality estimation for dense object detection. In *Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Nashville*, *TN*, *USA*, *20–25 June 2021*; IEEE: New York, 2021; pp. 11627–11636. doi: 10.1109/CVPR46437.2021.01146
23. Huang, L. H.; Zhao, X.; Huang, K. Q. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Trans. Pattern Anal. Mach. Intell., **2021**, *43*: 1562−1577. doi: 10.1109/TPAMI.2019.2957464
24. Fan, H.; Lin, L. T.; Yang, F.; *et al*. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Long Beach*, *CA*, *USA*, *15–20 June 2019*; IEEE: New York, 2019; pp. 5369–5378. doi: 10.1109/CVPR.2019.00552
25. Zhu, Z.; Wang, Q.; Li, B.; *et al*. Distractor-aware Siamese networks for visual object tracking. In *Proceedings of the 15th European Conference on Computer Vision*, *Munich*, *Germany*, *8–14 September 2018*; Springer: Berlin/Heidelberg, 2018; pp. 103–119. doi: 10.1007/978-3-030-01240-3_7
26. Wang, G. T.; Luo, C.; Xiong, Z. W.; *et al*. SPM-tracker: Series-parallel matching for real-time visual object tracking. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Long Beach*, *CA*, *USA*, *15–20 June 2019*; IEEE: New York, 2019, pp. 3638–3647. doi: 10.1109/CVPR.2019.00376
27. Xu, Y. D.; Wang, Z. Y.; Li, Z. X.; *et al*. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, *New York*, *NY*, *USA*, *7–12 February 2020*; AAAI: Palo Alto, 2020; pp. 12549–12556. doi: 10.1609/aaai.v34i07.6944
28. Danelljan, M.; Bhat, G.; Khan, F. S.; *et al*. ATOM: Accurate tracking by overlap maximization. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Long Beach*, *CA*, *USA*, *15–20 June 2019*; IEEE: New York, 2019; pp. 4655–4664. doi: 10.1109/CVPR.2019.00479
29. Hu, S.; Zhou, S. E.; Lu, J. B.; *et al*. Flexible dual-branch Siamese network: Learning location quality estimation and regression distribution for visual tracking. *IEEE Trans. Comput. Soc. Syst.* **2023**, in press. doi: 10.1109/TCSS.2023.3235649
30. Zheng, L. Y.; Tang, M.; Chen, Y. Y.; *et al*. Learning feature embeddings for discriminant model based tracking. arXiv: 1906.10414, 2020. doi: 10.48550/arXiv.1906.10414
31. Danelljan, M.; van Gool, L.; Timofte, R. Probabilistic regression for visual tracking. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Seattle*, *WA*, *USA*, *13–19 June 2020*; IEEE: New York, 2020; pp. 7181–7190. doi: 10.1109/CVPR42600.2020.00721
32. Zhang, J. P.; Dai, K. H.; Li, Z. W.; *et al*. Spatio-temporal matching for Siamese visual tracking. Neurocomputing, **2023**, *522*: 73−88. doi: 10.1016/j.neucom.2022.11.093
33. Kim, M.; Lee, S.; Ok, J.; *et al*. Towards sequence-level training for visual tracking. In *Proceedings of the 17th European Conference on Computer Vision*, *Tel Aviv*, *Israel*, *23–27 October*, *2022*; Springer: Berlin/Heidelberg, 2022, pp. 534–551. doi: 10.1007/978-3-031-20047-2_31
34. L i, J. F.; Li, B.; Ding, G. D.;, *et al*. Siamese global location-aware network for visual object tracking. Int. J. Mach. Learn. Cybern., **2023**, *14*: 3607−3620. doi: 10.1007/s13042-023-01853-2
35. Dai, K. N.; Zhang, Y. H.; Wang, D.; *et al*. High-performance long-term tracking with meta-updater. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Seattle*, *WA*, *USA*, *13–19 June 2020*; IEEE: New York, 2020; pp. 6297–6306. doi: 10.1109/CVPR42600.2020.00633
36. Huang, L. H.; Zhao, X.; Huang, K. Q. GlobalTrack: A simple and strong baseline for long-term tracking. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, *New York*, *NY*, *USA*, *7–12 February 2020*; AAAI: Palo Alto, 2019; pp. 11037–11044. doi: 10.1609/aaai.v34i07.6758
37. Danelljan, M.; Bhat, G.; Khan, F. S.; *et al*. ECO: Efficient convolution operators for tracking. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, *Honolulu*, *HI*, *USA*, *21–26 July 2017*; IEEE: New York, 2017; pp. 6931–6939. doi: 10.1109/CVPR.2017.733
38. Zhang, Z. P.; Peng, H. W. Deeper and wider Siamese networks for real-time visual tracking. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Long Beach*, *CA*, *USA*, *15–20 June 2019*; IEEE: New York, 2019; pp. 4586–4595. doi: 10.1109/CVPR.2019.00472
39. Song, Y. B.; Ma, C.; Wu, X. H.; *et al*. VITAL: VIsual tracking via adversarial learning. In *Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Salt Lake City*, *UT*, *USA*, *18–23 June 2018*; IEEE: New York, 2018; pp. 8990–8999. doi: 10.1109/CVPR.2018.00937
40. Fan, H.; Ling, H. B. Siamese cascaded region proposal networks for real-time visual tracking. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, *Long Beach*, *CA*, *USA*, *15–20 June 2019*; IEEE: New York, 2018; pp. 7944–7953. doi: 10.1109/CVPR.2019.00814
41. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, *Las Vegas*, *NV*, *USA*, *27–30 June 2016*; IEEE: New York, 2016; pp. 4293–4302. doi: 10.1109/CVPR.2016.465
42. Xu, T. Y.; Feng, Z. H.; Wu, X. J.; *et al*. Joint group feature selection and discriminative filter learning for robust visual object tracking. In *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision*, *Seoul*, *Korea* (*South*), *27 October 2019–02 November 2019*; IEEE: New York, 2019; pp. 7949–7959. doi: 10.1109/ICCV.2019.00804
43. Hu, W. M.; Wang, Q.; Zhang, L.; *et al*. SiamMask: A framework for fast online object tracking and segmentation. IEEE Trans. Pattern Anal. Mach. Intell., **2023**, *45*: 3072−3089. doi: 10.1109/TPAMI.2022.3172932

Citation: Hu, S.; Lu, J.; Zhou, S. Learning Regression Distribution: Information Diffusion from Template to Search for Visual Object Tracking. *International Journal of Network Dynamics and Intelligence*. 2024, 3(1), 100006. doi: 10.53941/ijndi.2024.100006