Graph Attention Transformer Network for Robust Visual Tracking

Libo Wang[1], Si Chen[1(✉)], Zhen Wang[2], Da-Han Wang[1], and Shunzhi Zhu[1]

[1] Fujian Key Laboratory of Pattern Recognition and Image Understanding, School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
wanglibo@stu.xmut.edu.cn, chensi@t.xmut.edu.cn, {wangdh,szzhu}@xmut.edu.cn

[2] School of Computer Science, Faculty of Engineering, The University of Sydney, Darlington, NSW 2008, Australia
zwan4121@uni.sydney.edu.au

Abstract. Visual tracking aims to estimate the state of an arbitrary object in a video frame only when the bounding box is given in the first frame. However, the existing trackers still struggle to adapt to complex environments due to the lack of adaptive appearance features. In this paper, we propose a graph attention transformer network, termed GATransT, to improve the robustness of visual tracking. Specifically, we design an adaptive graph attention module to enrich the embedding information extracted by the transformer backbone, which establishes the part-to-part correspondences between the template and search nodes. Extensive experimental results demonstrate that the proposed tracker outperforms the state-of-the-art methods on five challenging datasets, including OTB100, UAV123, LaSOT, GOT-10k, and TrackingNet.

Keywords: Visual tracking · Graph attention · Transformer

1 Introduction

Visual tracking plays a pivotal role in computer vision, aiming to estimate the state of an arbitrary object in a video frame according to the given initial target box. In recent years, object tracking has broad applications in intelligent traffic, video monitoring, and other fields. However, the performance of the existing trackers are influenced by various challenging factors, including illumination variation, deformation, motion blur, and background clutter.

Current mainstream trackers include Siamese-based trackers and transformer-based trackers, which have achieved good results in terms of efficiency and accuracy. Siamese-based trackers [1,13] utilize the cross-correlation for embedding information between the template and search branches. Transformer-based trackers [3,21,29] draw on the global and dynamic modeling capabilities to establish a long-distance correlation between the extracted template and search features.

M. Tanveer et al. (Eds.): ICONIP 2022, CCIS 1791, pp. 165–176, 2023.
https://doi.org/10.1007/978-981-99-1639-9_14

For example, STARK [29] proposes an encoder-decoder transformer architecture to model the global spatio-temporal feature dependencies between the target object and the search region.

Despite their great success, there are still some indispensable drawbacks. The transformer-based trackers can calculate the global and rich contextual interdependence between the template and the search region. However, the extracted features lack the part-level embedding information, resulting in the difficulty of adaptation to complex tracking scenarios. In addition, the template features extracted by the traditional trackers may contain too much redundant information, which will accumulate the tracking errors.

To solve the above two points, inspired by the graph attention network and the transformer, we propose a novel end-to-end graph attention transformer tracker GATransT that introduces the graph attention into the transformer-based tracker and establishes the local topological correspondences for the extracted features. We first utilize a transformer as a feature extraction network, which can obtain more semantic information through self-attention and cross-attention of the template and search region features. Next, we use the graph attention mechanism to propagate target information from the template to search region features. To reduce the interference of redundant template information and obtain more accurate tracking results, we employ an adaptive graph attention module to establish the correspondences between initial template nodes, dynamic template nodes and search nodes. In addition, we use the FocusedDropout operation to make the network focus on the target object, thus improving the tracking performance. As shown in Fig. 1, compared with the state-of-the-art trackers, our method can successfully track the target in cases of a similar object, background clutter, and partial occlusion. Finally, we evaluate the different trackers on public tracking benchmarks, including OTB100 [26], UAV123 [18], LaSOT [9], GOT-10k [12], and TrackingNet [19]. The experimental results show that the proposed tracker can outperform the competing trackers significantly.

The main contributions of this work can be summarized as follows:

- An end-to-end transformer-based graph attention tracking framework is proposed. To the best of our knowledge, this is the first work to introduce the graph attention into transformer for extracting the robust feature embedding information of the target.
- We employ an adaptive graph attention module to establish part-to-part correspondences by aggregating initial template nodes, dynamic template nodes, and search nodes to obtain robust adaptive features.
- Comprehensive experiments demonstrate the excellent performance of our method compared with the state-of-the-art trackers on the five challenging benchmarks.

2 Related Work

2.1 Visual Tracking

The current popular tracking paradigm contains the three main stages, i.e., feature extraction, feature fusion, and prediction. Most researchers focus on the

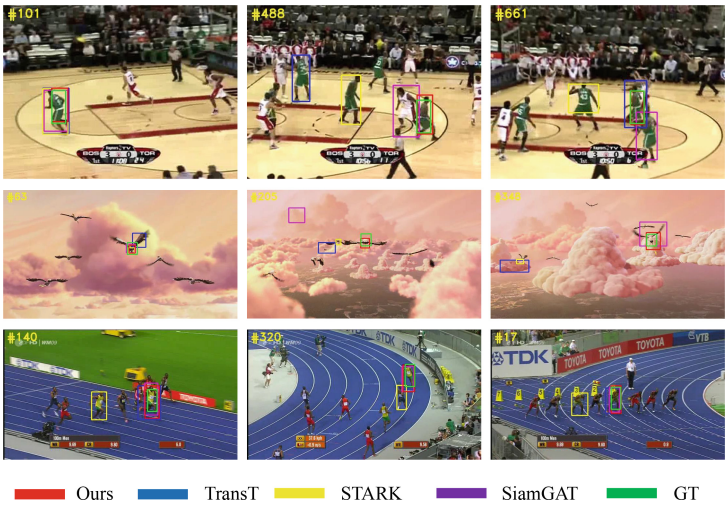

Fig. 1. Visualization comparison of the tracking results with TransT [3], STARK [29], and SiamGAT [11], where GT represents the ground-truth of object tracking. From top to bottom rows indicate the Basketball, Bird1, and Bolt sequences on the OTB100 dataset, respectively.

previous feature learning phase. Generally, almost all previous methods use CNN as the feature extraction network and recent works [6,22,24] also use transformer as the backbone. Regarding the critical feature fusion stage, the previous Siamese trackers often use a cross-correlation operation to obtain the fused response map of the template and search branches. And some online trackers learn the target model to distinguish the target foreground and background. Recently, several works [2,3,6,11,23] use the attention operation as a feature fusion method and also achieve the good performance. This paper will main focus on how to effectively introduce graph attention to transformer tracking.

2.2 Attention for Tracking

Attention mechanisms are often used in the visual tracking methods for feature fusion. On the one hand, the self-attention and cross-attention of the transformer are introduced as a module to improve the learning of long-range dependencies between the template and search branches. For example, TransT [3] introduces the attention operations to the transformer which replace the previous correlation operation to obtain the valuable feature maps. Mixformer [6] uses a transformer-based mixed attention backbone to extract more discriminative features and generate extensive interactions between the templates and search branches. On the other hand, graph attention is also applied in object tracking. SiamGAT [11] establishes the part-to-part correspondences between the target and the search region with a complete bipartite graph and propagates

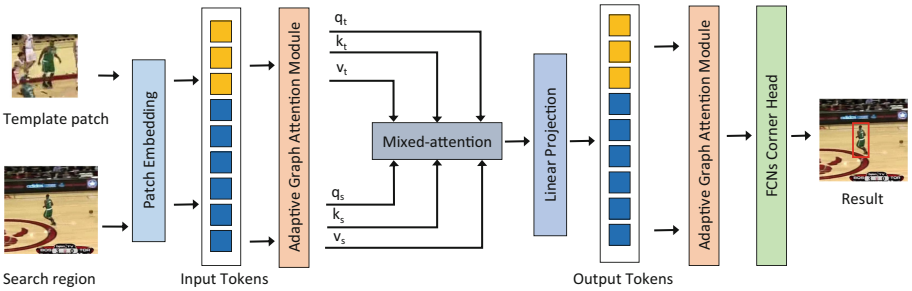

Fig. 2. Overview of the proposed transformer tracking framework based on the adaptive graph attention module.

target information from the template to the search feature. As in GraphFormers [30], it can capture and integrate the textual graph representation by making GNNs nested alongside each transformer layer of the pre-trained language model. Inspired by [30], we take advantage of the graph attention and transformer to obtain more robust adaptive features for visual tracking.

3 Proposed Method

3.1 Overview

In this section, we propose an effective graph attention transformer network GATransT for visual tracking, as shown in Fig. 2. The GATransT mainly contains the three components in the tracking framework, including a transformer-based backbone, a graph attention-based feature integration module, and a corner-based prediction head. In this framework, the adaptive graph attention module we designed enriches the embedding information extracted by the transformer backbone.

3.2 Transformer-Based Feature Extraction

Most previous trackers have adopted deep convolutional neural networks as feature extraction networks, such as AlexNet, ResNet, GoogleNet, etc. Although the effective feature extraction performance has been achieved, the extracted feature and semantic information are still not compact and rich enough. Inspired by Mixformer [6], we refer to the mixed attention network as the backbone, which can establish long-distance associations between the target template and search region to obtain richer feature representations. Since the transformer lacks the processing of part-level feature information, we design an adaptive graph attention module (Sect. 3.3 for more details).

In the process of feature extraction, we first convert the input template and search feature vector into tokens through patch embedding. Next, the convolution projection operation obtains the query, key, and value of the template and search features. This process can be formulated as

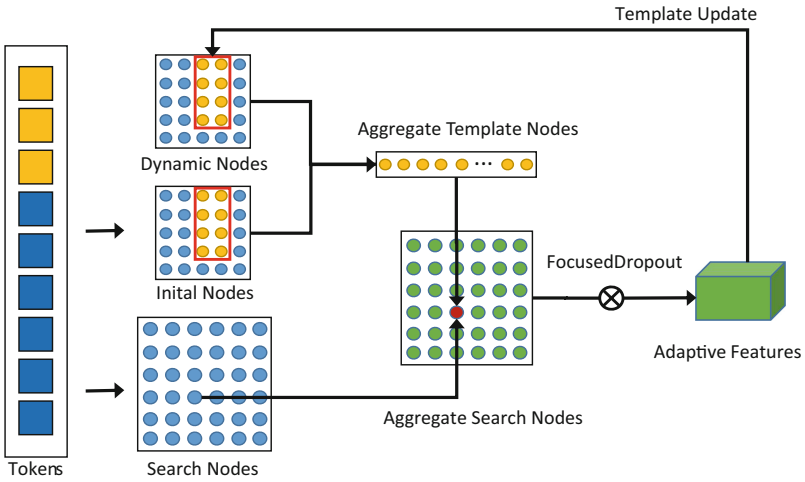

Fig. 3. Architecture of the adaptive graph attention module.

$$template^{q/k/v} = Flatten(Conv2d(Reshape2D(template), k\_s)), \tag{1}$$

$$search^{q/k/v} = Flatten(Conv2d(Reshape2D(search), k\_s)), \tag{2}$$

where $template^{q/k/v}$ and $search^{q/k/v}$ are the Q/K/V matrices obtained by the convolution projection of the template and search input token, respectively. $Conv2d$ is a depth-wise separable convolution, and k_s refers to the convolution kernel size.

Then we perform a mixed attention operation on the obtained queries, keys, and values. We use q_t, k_t, and v_t to represent the target, as well as q_s, k_s, and v_s to denote the search region. In this framework, the mixed attention is defined as

$$v_m = Concat(v_t, v_s), \tag{3}$$

$$Attn_t = Softmax(q_t k_t^T / \sqrt{d}) v_t, \tag{4}$$

$$Attn_s = Softmax(q_s k_m^T / \sqrt{d}) v_m, \tag{5}$$

where v_m is the value after concatenating the template and the search region; d represents the dimension of the key; $Attn_t$ and $Attn_s$ are the attention maps of the target and the search region, respectively. Finally, the target token and the search token are concatenated by a linear projection to get the output token.

3.3 Adaptive Graph Attention Module

Most existing trackers utilize cross-correlation operations or self-attention to perform feature fusion, which might lose semantic and part-level embedding

information. Inspired by the graph attention tracking, we establish part-to-part correspondences between the template and search region features extracted by the transformer backbone. These are achieved by aggregating initial template nodes, dynamic template nodes, and search nodes to obtain more robust adaptive features. As shown in Fig. 3, given two patches of the template and search region, we convert the respective tokens obtained through patch embedding into $h*w*c$ nodes to generate graphs, where h, w, and c represent the height, width, and channel of the feature, respectively.

In order to adaptively learn the feature representation between nodes, we calculate the correlation score between nodes by inner product to express the similarity of two nodes. Among them, to eliminate the background redundant information of the template, initial template nodes and dynamic template nodes are performed for graph attention to obtain more accurate template information. If the set update threshold is reached, the obtained adaptive features whose prediction results exceed a particular confidence score are used to update the dynamic template nodes. In addition, we perform softmax normalization to calculate the correlation scores α_{ij} so as to balance the amount of information as follows:

$$\alpha_t = Softmax((W_t p_t)^T (W_t' p_t')), \tag{6}$$

$$\alpha_{ij} = Softmax((W_s p_s)^T \alpha_t^j), \tag{7}$$

where W_t, W_t', W_s are the linear transformation matrix of initial template, dynamic template and search feature, respectively; p_t, p_t' and p_s refer to the node feature vectors of the template, the dynamic template and the search region, respectively.

In Eq. 7, the α_{ij} obtained by the above formula can be viewed as the attention given to the search graph node i according to the information of the template node j. Then all nodes in the template are propagated to the i-th node in the search area to calculate the aggregate feature representation of this node, which is written as

$$v_i = \sum_{j \in V_t} \alpha_{ij} W_v p_t^j, \tag{8}$$

where W_v represents the linear transformation matrix of the original feature; V_t represents the node set of template features; p_t^j refers to the template feature vector of node j. Finally, the aggregated features and the original features are combined to obtain a more robust feature representation as follows.

$$\hat{p_s^i} = Relu(Concat(v_i, W_v p_s^i)), \tag{9}$$

where p_s^i refers to the search region feature vector of node i. In addition, we refer to the FocusedDrop [27] operation on the aggregation node features after graph attention to obtain adaptive features that can focus on more robust target appearance features in the following formula:

$$P_s^i = FocusedDrop(\hat{p_s^i}, rate), \tag{10}$$

where the *rate* represents a participation rate.

4 Experiments

This section first describes the implementation details of our tracker. Then we analyze the influence of the main components in the proposed method. Finally, we compare the performance of our tracker and the state-of-the-art trackers on the OTB100 [26], UAV123 [18], LaSOT [9], GOT-10k [12], and TrackingNet [19] datasets.

4.1 Implementation Details

The proposed method is performed based on the deep learning framework PyTorch and implemented in an experimental environment of Intel-i7 CPU (32 GB RAM) and GeForce RTX TITAN (24 GB) with an average speed of about 11FPS. We compare our tracker with several state-of-the-art trackers on four public datasets and use one-pass evaluation (OPE) with precision and success plots on the challenging video sequences.

Training. We use the train splits of LaSOT [9], GOT-10K [12], COCO2017 [16], and TrackingNet [19] for offline training. The training strategy refers to Mixformer [6]. The entire training process is single-stage without too much parameter tuning and post-processing. Based on the original model, we added the proposed adaptive graph attention module to continue training. After 200 epochs of training, each tracking dataset has a certain effect. We train our tracker by using the ADAM optimizer and the weight decay of 0.0001. The learning rate is initialized as 1e−4. The sizes of search images and templates are 320 × 320 pixels and 128 × 128 pixels, respectively. For data augmentation strategies, we use horizontal flip and rotation to increase the amount of training data. We use the GIoU loss and the L1 loss for training loss with the weights of 2.0 and 5.0, respectively.

Inference. We take the initial template, multiple dynamic online templates, and the search area as the input of the tracker to generate the target bounding box and confidence scores. In this case, the dynamic template nodes of the adaptive graph attention module are updated only when the set update interval is reached, and the one with the highest confidence score is selected.

4.2 Ablation Study

To verify the effectiveness of each module of the proposed method, i.e., backbone, feature fusion, head, we conduct a detailed study of the different components on the LaSOT dataset. We use the STARK algorithm that removes temporal information as the baseline. The details of all the competing variants and the ablation results are listed in Table 1.

We design five different combinations for the three main components of backbone, feature fusion, and head. As shown in Table 1, we have several vital observations on the five different experimental settings. Firstly, the experimental setting #1 uses resnet-50 as the backbone, encoder-decoder as the feature fusion

Table 1. The ablation study of the main components of the proposed method on the LaSOT dataset.

Setting	Backbone	Feature Fusion	Head	AUC Score (%)
#1	RestNet-50	Encoder-Decoder	Corner	66.8
#2	Transformer	Graph	Corner	67.1
#3	Transformer	Graph+DTN	Corner	67.4
#4 (ours)	Transformer	Graph+DTN+FD	Corner	**67.5**
#5	Transformer	Graph+DTN+FD	Query	67.3

method, and corner as the feature prediction head. By comparing #1 and #2 in Table 1, we replace the backbone and feature fusion with transformer and graph attention, respectively, and the AUC score is improved by 0.3%. We introduce DTN (Dynamic Template Node) to the graph attention feature fusion method in #2, and the AUC score is improved by 0.3%. Then we add the FD (Focused-Dropout) operation based on #3, and the AUC score is increased by 0.1%. Finally, we compare the feature prediction head and find that the corner head is better than the query head. Overall, the main components of our proposed tracker demonstrate the effectiveness and exhibit excellent performance on the LaSOT dataset.

4.3 Comparisons with State-of-the-Art Trackers

In this section, we compare the GATransT with other advanced trackers on five challenge datasets, i.e., OTB100, UAV123, LaSOT, GOT-10k, and TrackingNet datasets.

OTB100. The OTB100 [26] dataset is composed of 100 video sequences, which include 11 challenge attributes. Several state-of-the-art trackers are compared in the experiments, including STARK-S [29], SiamRPN [14], GradNet [15], Deep-SRDCF [7], SiamDW [33], and SiamFC [1]. Figure 4 reports precision plots and success plots according to the one-pass evaluation (OPE) on the OTB100 dataset. The representative precision score is reported when the threshold is 20 in the legend of Fig. 4 (left). In Fig. 4 (right), when the overlap between the tracking result and the ground truth is greater than 0.5, the tracking is considered successful. We can see from Fig. 4 that GATransT has achieved the highest performance on the OTB100 dataset with the precision score of 88.8% and the AUC score of 68.1%, respectively. It is worth mentioning that compared with the STARK-S based transformer, the accuracy and AUC score of the proposed tracker are higher 0.6% and 0.8% on OTB100, respectively.

UAV123. The UAV123 [18] dataset contains 123 short-term video sequences and all sequences are fully annotated with upright bounding boxes. The UAV

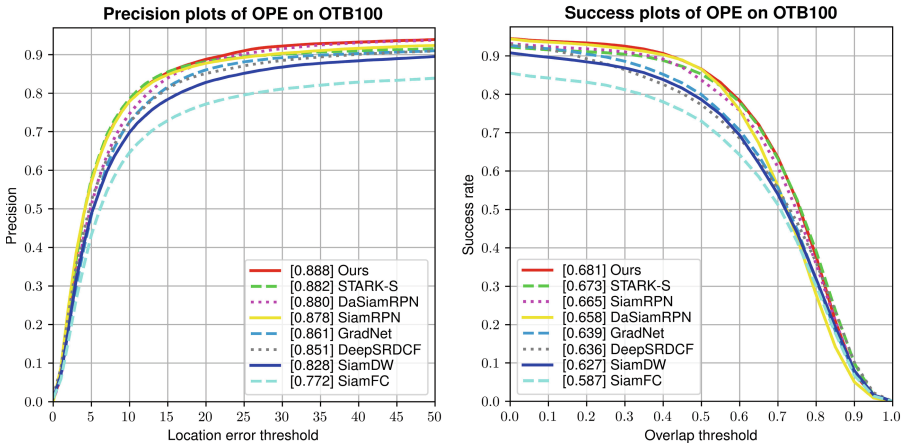

Fig. 4. Precision and success plots on the OTB100 dataset using the one-pass evaluation (OPE).

Table 2. Comparions with state-of-the-art trackers on the UAV123 dataset.

	CGACD [8]	SiamGAT [11]	SiamRCNN [20]	FCOT [4]	TREG [5]	STARK-S [29]	Ours
AUC (%)	63.3	64.6	64.9	65.6	66.9	67.2	**68.2**
Prec. (%)	83.3	84.3	83.4	87.3	88.4	88.5	**89.2**

dataset has the more challenging attributes than the OTB dataset, such as aspect ratio change, full occlusion, partial occlusion, and similar object. Table 2 reports the area under curve (AUC) scores and precision score values [25] compared with SiamFC [1], SiamRPN++ [13], CGACD [8], SiamGAT [11], SiamRCNN [20], FCOT [4], TREG [5], and STARK-S [29] on the UAV123 datasets. Among the competing tracking algorithms, our tracker works better than STARK-S in both AUC score and precision score due to the effective adaptive graph attention module to be used. Specifically, the AUC and precision scores of the GATransT are 68.2% and 89.2% on UAV123 respectively.

LaSOT/GOT-10k/TrackingNet. LaSOT [9] is a large-scale dataset for long-term tracking, which contains 280 videos with an average length of 2448 frames in the test set. GOT-10K [12] is a large-scale benchmark with over 10000 video segments and has 180 segments for the test set. TrackingNet [19] is a large-scale short-term dataset that contains 511 test sequences without publicly available ground truth. We evaluate the GATransT on the above three datasets, respectively. The compared state-of-the-art trackers include SiamRPN++ [13], SiamFC++ [28], D3S [17], Ocean [34], SiamGAT [11], DTT [31], STMTracker [10], SiamRCNN [20], AutoMatch [32], TrDiMP [21], and STARK-S [29]. From Table 3, our tracker shows excellent performance on three large-scale benchmarks, i.e., LaSOT, GOT-10k, and TrackingNet.

Table 3. Comparisons with state-of-the-art trackers on LaSOT, GOT-10k, and TrackingNet.

Tracker	LaSOT			GOT-10k			TrackingNet		
	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$	AUC	P_{Norm}	P
SiamRPN++ [13]	49.6	56.9	49.1	51.7	61.6	32.5	73.3	80.0	69.4
SiamFC++ [28]	54.4	62.3	54.7	59.5	69.5	47.9	75.4	80.0	70.5
D3S [17]	–	–	–	59.7	67.6	46.2	72.8	76.8	66.4
Ocean [34]	56.0	65.1	56.6	61.1	72.1	47.3	–	–	–
SiamGAT [11]	53.9	63.3	53.0	62.7	74.3	48.8	–	–	–
DTT [31]	60.1	–	–	63.4	74.9	51.4	79.6	85.0	78.9
STMTracker [10]	60.6	69.3	63.3	64.2	73.7	57.5	80.3	85.1	76.7
SiamRCNN [20]	64.8	72.2	–	64.9	72.8	59.7	81.2	85.4	80.0
AutoMatch [32]	58.2	–	59.9	65.2	76.6	54.3	76.0	–	72.6
TrDiMP [21]	63.9	–	61.4	67.1	**77.7**	58.3	78.4	83.3	73.1
STARK-S [29]	66.8	76.3	71.3	67.2	76.1	61.2	80.2	85.0	77.6
Ours	**67.5**	**76.9**	**72.5**	**67.2**	76.7	**62.9**	**80.6**	**85.1**	**77.8**

5 Conclusion

In this paper, we propose a novel graph attention transformer network for visual object tracking. This network leverages an adaptive graph attention to enrich long-distance correlation features extracted by the transformer backbone. The employed adaptive graph attention module can acquire robust target appearance features by establishing part-to-part correspondences between the initial template, dynamic template, and search nodes, thus adapting to complex tracking scenarios. The experimental results show that the proposed tracker can outperform the competing trackers significantly on five public tracking benchmarks, including OTB100, UAV123, LaSOT, GOT-10k, and TrackingNet.

Acknowledgement. This work was supported in part by the Natural Science Foundation of Fujian Province of China (Nos. 2021J011185 and 2021H6035); the Youth Innovation Foundation of Xiamen City of Fujian Province (No. 3502Z20206068); the Joint Funds of 5th Round of Health and Education Research Program of Fujian Province (No. 2019-WJ-41); and the Science and Technology Planning Project of Fujian Province (No. 2020H0023).

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 850–865. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_56

2. Chen, S., Wang, L., Wang, Z., Yan, Y., Wang, D.H., Zhu, S.: Learning meta-adversarial features via multi-stage adaptation network for robust visual object tracking. Neurocomputing **491**, 365–381 (2022)
3. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8126–8135 (2021)
4. Cui, Y., Jiang, C., Wang, L., Wu, G.: Fully convolutional online tracking. arXiv preprint arXiv:2004.07109 (2020)
5. Cui, Y., Jiang, C., Wang, L., Wu, G.: Target transformed regression for accurate tracking. arXiv preprint arXiv:2104.00403 (2021)
6. Cui, Y., Jiang, C., Wang, L., Wu, G.: MixFormer: end-to-end tracking with iterative mixed attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
7. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 58–66 (2015)
8. Du, F., Liu, P., Zhao, W., Tang, X.: Correlation-guided attention for corner detection based visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6835–6844 (2020)
9. Fan, H., et al.: LaSOT: a high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5374–5383 (2019)
10. Fu, Z., Liu, Q., Fu, Z., Wang, Y.: STMTrack: template-free visual tracking with space-time memory networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13774–13783 (2021)
11. Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., Shen, C.: Graph attention tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9543–9552 (2021)
12. 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)
13. 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 Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4282–4291 (2019)
14. 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 (CVPR), pp. 8971–8980 (2018)
15. Li, P., Chen, B., Ouyang, W., Wang, D., Yang, X., Lu, H.: GradNet: gradient-guided network for visual object tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 6162–6171 (2019)
16. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
17. Lukezic, A., Matas, J., Kristan, M.: D3S - a discriminative single shot segmentation tracker. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7131–7140 (2020)
18. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for UAV tracking. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 445–461. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_27

19. Müller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: TrackingNet: a large-scale dataset and benchmark for object tracking in the wild. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 310–327. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_19
20. Voigtlaender, P., Luiten, J., Torr, P.H.S., Leibe, B.: Siam R-CNN: visual tracking by re-detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6577–6587 (2020)
21. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: exploiting temporal context for robust visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
22. Wang, Z., Liu, L., Duan, Y., Kong, Y., Tao, D.: Continual learning with lifelong vision transformer. In: CVPR, pp. 171–181 (2022)
23. Wang, Z., Liu, L., Duan, Y., Tao, D.: SIN: semantic inference network for few-shot streaming label learning. IEEE Trans. Neural Netw. Learn. Syst. 1–14 (2022)
24. Wang, Z., Liu, L., Kong, Y., Guo, J., Tao, D.: Online continual learning with contrastive vision transformer. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13680, pp. 631–650. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20044-1_36
25. Wang, Z., Liu, L., Tao, D.: Deep streaming label learning. In: International Conference on Machine Learning (ICML), vol. 119, pp. 9963–9972 (2020)
26. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
27. Xie, T., Liu, M., Deng, J., Cheng, X., Wang, X., Liu, M.: Focuseddropout for convolutional neural network. arXiv preprint arXiv:2103.15425 (2021)
28. Xu, Y., Wang, Z., Li, Z., Ye, Y., Yu, G.: SiamFC++: towards robust and accurate visual tracking with target estimation guidelines. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 12549–12556. AAAI Press (2020)
29. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 10428–10437 (2021)
30. Yang, J., et al.: GraphFormers: GNN-nested transformers for representation learning on textual graph. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 28798–28810 (2021)
31. Yu, B., et al.: High-performance discriminative tracking with transformers. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 9836–9845 (2021)
32. Zhang, Z., Liu, Y., Wang, X., Li, B., Hu, W.: Learn to match: automatic matching network design for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 13319–13328 (2021)
33. Zhang, Z., Peng, H.: Deeper and wider siamese networks for real-time visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4591–4600 (2019)
34. Zhang, Z., Peng, H., Fu, J., Li, B., Hu, W.: Ocean: object-aware anchor-free tracking. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 771–787. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_46