**QUEEN'S
UNIVERSITY
BELFAST**

VTT: Long-term Visual Tracking with Transformers

Bian, T., Hua, Y., Song, T., Xue, Z., Ma, R., Robertson, N., & Guan, H. (2021). VTT: Long-term Visual Tracking with Transformers. In *International Conference on Pattern Recognition (ICPR)* (International Conference on Pattern Recognition (ICPR): Proceedings). Institute of Electrical and Electronics Engineers Inc..

Published in:
International Conference on Pattern Recognition (ICPR)

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights
Copyright 2020 IEEE. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

VTT: Long-term Visual Tracking with Transformers

Tianling Bian*, Yang Hua[†], Tao Song*, Zhengui Xue*, Ruhui Ma*, Neil Robertson[†] and Haibing Guan*

*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

{bian_tianling, songt333, zhenguixue, ruhuima, hbguan}@sjtu.edu.cn

[†]School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, UK

{Y.Hua, N.Robertson}@qub.ac.uk

Abstract—Long-term visual tracking is a challenging problem. State-of-the-art long-term trackers, e.g., GlobalTrack, utilize region proposal networks (RPNs) to generate target proposals. However, the performance of the trackers is affected by occlusions and large scale or ratio variations. To address these issues, in this paper, we are the first to propose a novel architecture with transformers for long-term visual tracking. Specifically, the proposed Visual Tracking Transformer (VTT) utilizes a transformer encoder-decoder architecture for aggregating global information to deal with occlusion and large scale or ratio variation. Furthermore, it also shows better discriminative power against instance-level distractors without the need for extra labeling and hard-sample mining. We conduct extensive experiments on three large-scale long-term tracking datasets and have achieved state-of-the-art performance.

I. INTRODUCTION

Long-term visual tracking, first proposed in [1], is a challenging sub-problem in visual object tracking, where videos are much longer and the target may constantly disappear, be occluded, change drastically in scales and aspect ratios, or be confused with similar distractors during tracking. Long-term tracking, by definition, is closer to real-life tracking scenarios, hence it has a great potential in many real-life applications, including smart surveillance [2], UAV monitoring [3] and autonomous driving [4].

Many existing approaches have been proposed for long-term tracking, such as TLD [1], SPLT [5], DaSiam_LT [6], EBT [7] and GlobalTrack [8]. GlobalTrack, specifically, has achieved great progress with a Faster-RCNN-like two-stage architecture and a pure global-search paradigm. However, its performance degrades when it faces challenging scenarios. Specifically, when occlusion, large scale or ratio variation and instance-level distractors appear, GlobalTrack fails to track the correct target because of limited capability to aggregate global information and lack of discriminative power. Some typical failure cases of GlobalTrack are shown in Fig. 1.

Inspired by the success of Transformers [9] and its recent applications in Computer Vision [10], [11], in this paper, we propose a Visual Tracking Transformer (VTT) armed with a transformer encoder-decoder architecture, which aggregates global information to benefit challenging scenarios for the long-term tracking. Specifically, the proposed tracker consists

Fig. 1. A comparison of VTT with GlobalTrack in challenging scenarios. The first row is a sample video where the target object is occluded. The second row is a sample video where the target undergoes drastic scale and aspect ratio change. The third row is a sample video where many instance-level distractors are present. When faced with occlusion, large scale or aspect ratio variation and instance-level distractors, GlobalTrack fails while VTT performs well in such cases. *Best viewed in color.*

of the following major components: a shared backbone for correlation feature extraction, a transformer for aggregating global information and a prediction feed-forward network (FFN) to perform bounding box regression and classification on transformer outputs. To the best of our knowledge, this work is the first to apply transformer architecture in the context of long-term tracking.

During tracking, VTT first flattens the extracted correlation feature map to construct a sequence of regions. Then it uses transformer encoder to aggregate global information, which models pair-wise relations between regions across long distances and benefits occlusion. It also shows better generalization across scales and ratios without adopting feature-pyramid networks (FPNs). Next, a transformer decoder decodes a fixed number of pre-trained queries into the same number of predictions, conditioned on the encoder memory. Afterwards, prediction FFN is used to regress relative bounding box coordinates and perform classification. The top-1 prediction is simply taken as the tracking result, which is the same way as GlobalTrack [8] does. To train transformer architecture in the context of long-term tracking, we propose a novel optimal assignment loss which takes into account all predictions without potentially dropping hard false-positive samples. Without the need for human labor to annotate distractors or complex

This work was supported in part by National Natural Science Foundation of China (NO. 61872234, 61732010, 61525204) and Shanghai Key Laboratory of Scalable Computing and Systems. Tao Song (songt333@sjtu.edu.cn) is the corresponding author.

hard-sample mining strategies, this makes our framework more data efficient during training and more discriminative.

The performance of VTT is verified on three large-scale long-term tracking benchmarks, including LaSOT [12], TLP [13] and OxUvA [14]. VTT outperforms GlobalTrack in all benchmarks and achieves state-of-the-art performance.

To conclude, the main contributions of this paper are threefold:

- It is the first to propose a novel long-term tracker, namely Visual Tracking Transformer(VTT), utilizing a transformer encoder-decoder architecture. A novel optimal assignment loss is designed for training VTT.
- The proposed architecture aggregates global information to benefit challenging scenarios including occlusion and scale or ratio variation. It also shows better discriminative power against instance-level distractors without extra labelling or hard-sample mining.
- Extensive experiments are conducted on three long-term tracking benchmarks. Without bells and whistles, our approach outperforms GlobalTrack in all datasets and achieves state-of-the-art results.

II. RELATED WORK

A. Long-term Tracking

Long-term tracking is the task of continuously locating an arbitrary object in a relatively long video [14]. Compared with short-term tracking, long-term tracking can reflect more practical performance of a tracker in the wild [12]. In long-term tracking, a target could undergo various difficult conditions. The most challenging ones include occlusions, drastic scale or ratio variations and distractors. This poses challenges to traditional tracking algorithms.

Occlusion is one of the greatest challenges for tracking. Many competitive short-term trackers, such as ECO [15] and SiamRPN++ [16], fails in occlusion. Other recent studies perform a global search to mitigate this issue. TLD [1], SPL [17], and DaSiam_LT [6] redetect the tracking target globally in case of tracking failure, while EBT [7] and GlobalTrack [8] search globally at every frame. However, they address occlusion in an indirect way to avoid its effect on performance. How to accurately track objects during partial and heavy occlusion has not been addressed.

To deal with drastic scale and ratio variation, many trackers such as ECO [15] and SiamRPN++ [16] utilize previous tracking results to crop a tracking window and resize it to a fixed size. These trackers only work when target trajectories are smooth. GlobalTrack performs a full image search using region proposal networks (RPNs) [18]. It utilizes ROI-Align [19] to extract features and adopts FPNs [20] to cover all possible sizes. But its performance on large scale objects is limited because of the long-tail distribution of object size in the training set.

Instance level distractor is also a challenge for long-term trackers. ATOM [21] trains a classification component online during tracking to improve discriminative power in the

presence of distractors. However, this leads to high risk of drifting in very long sequences. DaSiamRPN [6] introduces hard samples from the same category during training to learn more discriminative features. GlobalTrack [8] introduces a cross-query loss where it tries to locate different instances in the same image and average the loss. However, these approaches rely on categorical information or extra labelling of other instances in the same frame, which limits the amount of data they can utilize. They also break the premise of tracking, where only visual information of tracking target is available.

In terms of the tracking framework, our method shares similarity with GlobalTrack. We perform the global search for the template object in each frame, without relying on any spatial or temporal locality assumptions. However, our method differs from GlobalTrack in the following three aspects: (1) Our method formulates the task of locating desired object in the whole image as modeling long distance relations between all region pairs in that image and it uses a transformer encoder-decoder architecture to generate predictions. In contrast GlobalTrack reuses RPN and RCNN heads from Faster-RCNN. (2) Our method performs tracking on one level feature and it shows better performance than GlobalTrack, which uses feature pyramid networks. (3) Our model predicts a reasonable amount of predictions and our optimal assignment loss takes into account all the predictions. This allows us to locate instance-level distractors in an end-to-end way and improve discriminative ability without the need for cross-query loss or extra hard-sample mining strategies.

B. Transformers for Sequence to Sequence Prediction

Transformers [9] were first proposed for machine translation. One of the greatest advantages of transformers is the ability to compute global relations across long distances, making them more competitive than RNNs in very long sequences. Transformers are now gaining popularity over RNNs in sequence modeling tasks, such as natural language processing [9], [22], [23] and computer vision [10], [24], [11]. DETR [11] is the first to apply transformer architecture on detection. It uses transformers to simplify the surrogate tasks used in state-of-the-art detection methods and forms an end-to-end architecture. We take inspiration from DETR to build an end-to-end process using transformers, while aggregating global information to benefit challenging scenarios.

III. VISUAL TRACKING WITH TRANSFORMERS

In this work, we present Visual Tracking Transformer (VTT), a global instance search based tracker. Our method consists of four components: a siamese backbone for correlation feature extraction, a transformer encoder for aggregating global information, a transformer decoder to transform a fixed number of pre-trained target queries into the same number of predictions and a feed-forward network to convert decoder predictions and produce tracking results. An overview of our method is shown in Fig. 2.

In the following, we first introduce our VTT architecture in detail. Then we present the optimal assignment loss used

Fig. 2. The overall architecture of VTT. Siamese backbone is used to extract correlation feature. Transformer encoder encodes all pair-wise relations between all regions, using the whole image as context. Transformer decoder produces a fixed number of predictions given a pre-trained set of target-queries. The predictions are then converted to bounding box predictions and confidence scores with a prediction FFN. During tracking, we crop the given target as the template and take top-1 prediction as the tracking result. *Best viewed in color.*

in our approach. Finally, we describe the offline training and online tracking process using VTT.

A. VTT Architecture

a) Correlation Feature Extraction: Similar to SiamRPN, we use Siamese networks to extract features of template and search frames and encode their relationship using correlation. A Siamese network consists of two branches: a template branch that takes the template Z_{img} as input and a search branch that takes the current frame X_{img} as input. Both branches share the same architecture and parameters to extract features $Z_{feat} \in R^{C \times S \times S}$ and $X_{feat} \in R^{C \times H \times W}$. In order to embed information between the Z_{feat} and X_{feat} , the following function is first learned to transform Z_{feat} into a 1×1 kernel:

$$Z_{kern} = \varphi(Z_{feat}). \quad (1)$$

The function φ is implemented by a convolution layer with no activation. Z_{kern} contains the important visual features of the template patch. Then a correlation map X_{corr} is obtained by performing depth-wise correlation between Z_{feat} and X_{feat} , as follows

$$X_{corr} = X_{feat} \otimes Z_{kern}, \quad (2)$$

where $\otimes$ denotes a channel-by-channel correlation operation. We show this process in Fig. 3. Since Z_{kern} is a 1×1 kernel, the resulting X_{corr} has the same size as X_{feat} , and it contains the similarity between X_{feat} and Z_{kern} . We use a 1×1 convolution to reduce X_{corr} to a lower dimension correlation map $X'_{corr} \in R^{d \times H \times W}$. Since the transformer encoder requires a sequence as input, we reshape the two dimensional correlation map into one dimension, resulting in a sequence $S_{corr} \in R^{d \times H \times W}$.

b) Transformer Encoder: Transformer encoders use self-attention mechanism to aggregate global information to benefit occlusion and scale or ratio variation. It takes sequence S_{corr} as input and produces encoder memory F_{enc} as output. It consists of several identical encoder layers. Each encoder layer follows a standard architecture as described in [9] and consists of a multi-head self-attention module and an FFN with residual connections.

Fig. 3. Illustration of the process of obtaining correlation map X_{corr} . The template feature Z_{feat} is first converted into a 1×1 kernel Z_{kern} by φ . Then channel-by-channel correlation operation is performed between Z_{kern} and search image feature X_{feat} to get correlation feature X_{corr} . *Best viewed in color.*

Multi-head self-attention is composed of m self-attention heads to learn multiple distinct transforms in different feature spaces. The input sequence F is split channelwise into m sequences, $F'_i \in R^{d' \times n}, i \in \{1, \dots, m\}$, $d' = \frac{d}{m}$ and $n = H \times W$. The i^{th} self-attention head takes sequence F'_i as an input, and first obtains query, key and value by

$$Q = W_Q(F'_i + P), \quad (3)$$

$$K = W_K(F'_i + P), \quad (4)$$

$$V = W_V F'_i, \quad (5)$$

where the query Q , key K , and value V are linear transformations of sequence F'_i . $W_Q, W_K, W_V \in R^{d' \times d'}$. In practice, we use 1×1 convolution with d' input and output channels. P is a fixed positional encoding [10], [24] added to the input of each attention layer so that the spacial relationship of the two dimensional feature map is preserved after being collapsed into one dimension. Then it performs attention operation between Q, K, V by

$$\hat{F}'_i = \text{softmax} \left(\frac{Q^\top K}{\sqrt{d'}} \right) V, \quad (6)$$

where d' is a temperature parameter to control the intensity of attention. The multi-head self-attention output is simply the concatenation of m single self-attention heads:

$$\hat{F} = [\hat{F}'_1; \dots; \hat{F}'_m], \quad (7)$$

where $[\cdot; \cdot]$ denotes concatenation on the channel axis. $\hat{F}$ has the same dimension as F .

An FFN layer is added after the multi-head self-attention layer. In our implementation, we use two-layers of 1x1 convolutions with ReLU activations. We add residual connection, dropout, layernorm after multi-head attention and FFN:

$$F_{att} = \text{layernorm}(F + \text{dropout}(\hat{F})), \quad (8)$$

$$F_{out} = \text{layernorm}(F_{att} + \text{dropout}(\text{FFN}(F_{att}))), \quad (9)$$

where $F_{out} \in R^{d \times n}$ is the output of an encoder layer and it is used as input for each next encoder layer. The encoder memory F_{enc} is the output of the last encoder layer.

c) *Transformer Decoder*: To generate a set of N target predictions, we supply the decoder with encoder memory F_{enc} and a sequence of learned target query vectors $Q_{obj} \in R^{d \times N}$ to produce an output embedding $D \in R^{d \times N}$. Similar to the transformer encoder, the transformer decoder also consists of several decoder layers, and each of them contains a multi-head encoder-decoder attention layer and an FFN.

A multi-head encoder-decoder attention layer splits encoder memory F_{enc} and query sequence Q_{dec} channelwise into m sequences, respectively denoted by $F_{enc}^i \in R^{d' \times n}$, and $Q_{dec}^i \in R^{d' \times N}$, $i \in [1, \dots, m]$. Similar to Eq. 3 - 5, the query, key and value of the i^{th} encoder-decoder attention head is obtained by

$$Q_d = W_Q^d(Q_{dec}^i + Q_{obj}), \quad (10)$$

$$K_d = W_K^d(F_{enc}^i + P), \quad (11)$$

$$V_d = W_V^d F_{enc}^i. \quad (12)$$

The attention is then performed in the following way similar to Eq. 6:

$$\hat{F}_i^d = \text{softmax}\left(\frac{Q_d^\top K_d}{\sqrt{d'}}\right) V_d. \quad (13)$$

The output of each encoder-decoder attention head is concatenated, forming $\hat{F}^d$. We use residual connections around multi-head self-attention layer and FFN layer and apply dropout and layer norm in the same way as what we do with the transformer encoder. For the first decoder layer, $Q_{dec} = Q_{obj}$ and for the rest of the decoder layers Q_{dec} is the output of the previous decoder layer.

d) *Prediction feed-forward network*: We use a bounding box regression FFN and a classification FFN to compute object bounding box and score, respectively. The regression FFN consists of three linear layers with ReLU activation and a sigmoid layer. It predicts the relative center coordinates, height and width of the bounding box with respect to the image size. The classification FFN is a linear layer that predicts binary classification using a softmax function. The output is interpreted as the confidence of the respective bounding box prediction being the tracking target.

B. Optimal Assignment Loss

For each image pair, VTT produces a fixed number of N target predictions, denoted as $\hat{y}$. We first choose the optimal prediction and assign it to the ground truth target, then we calculate classification and bounding box regression loss according to the assignment result.

Similar to the training of RPN, the first step is to find the optimal prediction to assign to the ground truth. We design a simple assignment cost that takes into account both the score prediction and the bounding box prediction. Let b be the ground truth bounding box, $\hat{y}_i = (\hat{p}_i, \hat{b}_i)$ denote the i^{th} prediction, $\hat{p}_i \in [0, 1]$ be the confidence score and $b, \hat{b}_i \in [0, 1]^4$ be vectors that define the relative position of bounding box center and relative width and height with respect to the image size. The assignment cost is defined as:

$$\mathcal{C}_{\text{assign}}(\hat{y}_i) = -\hat{p}_i + \mathcal{L}_{\text{box}}(b, \hat{b}_i), \quad (14)$$

where $\mathcal{L}_{\text{box}}(\cdot, \cdot)$ is a combination of ℓ_1 loss and generalized IoU loss [25]:

$$\mathcal{L}_{\text{box}}(b, \hat{b}) = \alpha \cdot \mathcal{L}_{\text{giou}}(b, \hat{b}) + \beta \cdot \|b - \hat{b}\|_1, \quad (15)$$

where α, β are hyperparameters to balance $\mathcal{L}_{\text{giou}}$ and ℓ_1 loss.

The prediction with the lowest $\mathcal{C}_{\text{assign}}$ is assigned to ground truth. Let i_o denote its index. The labels for each prediction is set to

$$c_i = \begin{cases} 1, & i = i_o \\ 0, & i \neq i_o \end{cases}. \quad (16)$$

The classification loss is a binary cross entropy loss:

$$\mathcal{L}_{\text{bce}}(c, \hat{p}) = -c \log \hat{p} - (1 - c) \log(1 - \hat{p}). \quad (17)$$

The final training loss is then defined as a combination of $\mathcal{L}_{\text{bce}}$ and $\mathcal{L}_{\text{box}}$:

$$\mathcal{L}(y, \hat{y}) = \mathcal{L}_{\text{box}}(b, \hat{b}_{i_o}) + \gamma \cdot \sum_{i=1}^N \mathcal{L}_{\text{bce}}(c_i, \hat{p}_i), \quad (18)$$

where γ is hyperparameter to scale $\mathcal{L}_{\text{bce}}$.

C. Offline Training and End-to-End Hard Sample Mining

We randomly sample two frames from the same training video to construct a training pair. For each pair, we choose one co-exist object in both frames. For one frame, we crop a square patch according to the object center position and the longer edge of width and height and resize it to a fixed size as Z_{img} . For the other frames, we simply take the whole frame as X_{img} . We run one forward pass of the whole network, and compute the proposed loss using Eq. 18.

VTT mines hard samples in an end-to-end fashion. Unlike GlobalTrack, it only requires one instance that exists in both frames, without needing cross-query loss that runs the same frame multiple passes with different instances as templates. The reason is that hard samples are effectively false positive predictions. Since VTT only produces a reasonable number of predictions, the loss in Eq. 18 is calculated without any sampling, making efficient use of hard samples in the same frame. This makes our model more data-efficient and can easily utilize more data in the future. Experiments show that our model is more discriminative against instance-level distractors and it will be further discussed in Section IV-D.

Fig. 4. Precision and success plots of our method, in comparison with GlobalTrack and other state-of-the-art trackers on the LaSOT [12] test set. Compared with GlobalTrack, our method achieves absolute gains of 6.7% in precision score and 4.0% in success score. VTT also outperforms all other trackers by a large margin. *Best viewed in color.*

TABLE I
EXPERIMENT RESULT ON TLP[13]

(%)	SR _{0.5}	Success	Precision
LCT [26]	8.7	9.9	7.2
ECO [15]	21.9	20.2	21.2
ADNet [27]	22.1	22.3	20.3
MDNet [28]	42.3	37.0	38.4
MBMD [29]	48.1	-	-
SiamRPN [30]	51.5	-	-
ATOM [21]	47.5	-	-
SPLT [5]	52.7	-	-
GlobalTrack [8]	63.8	52.0	55.6
VTT (Ours)	66.3	52.2	57.6

D. Online Tracking

The tracking process is simple and straightforward. In the first frame, we crop the template patch in the same way as in training. The template patch is fixed throughout the tracking process with no updating. In each tracking frame, we run a forward pass of VTT with template patch and current frame as inputs. We take the top-1 prediction as the tracking result. We conduct no post-processing, e.g. trajectory refinement and penalization on large displacements, on the tracking result for a fair comparison with GlobalTrack, which also does not adopt any extra post-processing.

IV. EXPERIMENTS

A. Implementation details

We use ResNet-50 [32] as our backbone. The channel number of the backbone feature is $c = 2048$. The backbone of our model is initialized from DETR pretrained on COCO [33] dataset. Following GlobalTrack, we train our model on a combination of COCO [33], GOT-10k [34] train set and LaSOT [12] train set for a fair comparison. COCO is an image object detection dataset containing over 118k images and 80 object classes. GOT-10k and LaSOT are visual tracking datasets, where GOT-10k consists of 10,000 videos belonging to 563 object classes and LaSOT consists of 1,400 videos belonging to 70 object classes. For the COCO dataset, from each image, we randomly select one annotated object as the

TABLE II
EXPERIMENT RESULT ON OXUVA TEST SET

(%)	MaxGM	TPR	TNR
ECO-HC [15]	31.4	39.5	0.0
MDNet [28]	34.3	47.2	0.0
SINT [31]	32.6	42.6	0.0
EBT [7]	28.3	32.1	0.0
DaSiam_LT [6]	41.5	68.9	0.0
VTT (Ours) w/o presence	43.0	74.0	0.0
TLD [1]	43.1	20.8	89.5
LCT [26]	39.6	29.2	53.7
SiamFC+R [14]	45.4	42.7	48.1
MBMD [29]	54.4	60.9	48.5
SPLT [5]	62.2	49.8	77.6
GlobalTrack [8]	60.3	57.4	63.3
VTT (Ours)	61.6	61.7	61.5

Fig. 5. Left column: Visualization of decoder attention for example frames where the target is heavily occluded, in extreme scale variation and in extreme ratio variation, respectively. Decoder attention can focus on disconnected regions of the same object, making it robust against these challenging situations. Right: Precision plots on the LaSOT dataset for sequences with Full Occlusion, Scale Variation and Aspect Ratio Change attributes. VTT performs better than any other tracker under occlusion and scale or ratio variation. *Best viewed in color.*

template to compose a template-search pair. For GOT-10k and LaSOT datasets, we randomly sample pairs of frames from videos with one as the template patch and the other as the search frame. Scales, crops and horizontal flips are used for augmentation.

Fig. 6. We visualize self-attention from different encoder layers for a specific point in the feature map. Lower level encoder layers (left side) attend to scattered areas that are visually related, while higher level encoder layers (right side) can aggregate global information and attend to connected regions that belong to the tracking target. This example also shows the reasoning process of occlusion. Specifically, layer 5 attends to regions that are directly connected with the reference point, while layer 6 can attend to regions that are disconnected from the reference point because of partial occlusion. This reasoning process improves the robustness of VTT against occlusion. *Best viewed in color.*

TABLE III
ENCODER ABLATION

Encoder layers	Success	Δ_S	Precision	Δ_P
0	11.8	-43.9	2.4	-57.1
3	47.6	-8.1	45.8	-13.7
6	55.7	-	59.5	-

We train VTT using AdamW [35]. The backbone is pre-trained on COCO and is fixed during training. The weights for each loss described in Eq. 15 and 18 are $\alpha = 1$, $\beta = 5$ and $\gamma = 0.01$. We first train our network on COCO for 20 epochs with a learning rate set to 1×10^{-4} . Then we train it on a combination of COCO, GOT-10K and LaSOT, as described above, for another 100 epochs. The initial learning rate is set to 1×10^{-4} , and decays with a factor of 0.1 at epoch 80. We implement our method in Python using PyTorch. The training process takes seven days on eight RTX 2080Ti GPUs with two images per GPU, while online-tracking runs at 12fps on a single GPU.

B. Long-term Visual Object Tracking Evaluation

We evaluate our tracker on three large-scale long-term tracking benchmarks.

LaSOT is the largest long-term tracking dataset [12] for large scale long-term tracking. The test set contains 280 videos with an average length of 2448 frames. Such a large dataset introduces great challenges to tracking algorithms. We compare our tracker with GlobalTrack as well as other state-of-the-art trackers including ATOM [21], SiamRPN++ [16], MDNet [28], VITAL [36], SiamFC [37], DSiam [38] and ECO [15]. The results of GlobalTrack and ATO are obtained from the project website, while the others are provided by the official website of LaSOT. As shown in Fig. 4, our method outperforms GlobalTrack as well as other methods by a large margin, achieving absolute gains of 6.7% in the precision score and 4.0% in the success score. The results in such a large dataset demonstrate that our method using transformers has great advantages over traditional methods.

TABLE IV
DECODER ABLATION

Decoder layers	Success	Δ_S	Precision	Δ_P
1	54.0	-1.7	57.3	-2.2
3	55.3	-0.4	58.7	-0.8
6	55.7	-	59.5	-

TLP [13] is a large-scale long-term tracking benchmark containing 50 long videos, each with an average length of 13529 frames. Success rate ($SR_{0.5}$, under overlap threshold 0.5), success score and precision score are used as evaluation metrics. We compare our method with GlobalTrack [8], SPLT [5], ATOM [21], SiamRPN [30], MBMD [29], MDNet [28], ADNet [27], LCT [26], ECO [15]. The results are shown in Table I. Our method achieves a 2.5% absolute gain in $SR_{0.5}$ compared to GlobalTrack and it significantly outperforms all other methods. The results show that our approach is competitive in very long videos.

OxUvA [14] is a large scale long-term tracking benchmark. Its test set contains 166 videos with an average length of 3293 frames. The ground truth of the test set is withheld. The tracker performance is measured with true positive rate (TPR), true negative rate (TNR) and maximum geometric mean of TPR and TNR (MaxGM). The results are shown in Table II. There are two styles to estimate tracker performance. One is to predict a bounding box for each frame and compare TPR. VTT achieves 74.0 in TPR and outperforms the reported best tracker DaSiam_LT [6] by an absolute gain of 5.1%. Another one is to estimate the target presence for each frame. We simply set a threshold for the top-1 score of the final prediction. The frames where the top-1 scores exceed $\mu = 0.97$ are considered as *target presence*. Compared with the reported best tracker SiamFC-R, VTT-PD achieves 16.2% absolute gain in terms of MaxGM. It also outperforms GlobalTrack by 1.3% in MaxGM and achieves competitive results against other state-of-the-art methods. The results show that VTT achieves strong performance in long-term tracking.

C. Ablation Study

In our ablation study, we show how each component of transformer influences the final performance. We choose VTT with ResNet-50 backbone, 6 encoder layers, 6 decoder layers and hidden size 256. The models are evaluated on the test set of LaSOT.

1) *Encoder Layers*: The importance of aggregating global information is evaluated by experimenting with the number of encoder layers. The results are shown in Table III. Without encoder layers, the success and precision score drops by 43.9 points and 57.1 points, respectively. This indicates that the global aggregation capability achieved with encoder layers is essential in our architecture. We observe improvement in performance when the number of encoder layers increases. Fig. 6 visualizes the attention weights for a specific reference point from the current target. Although each single encoder

Fig. 7. (a) Average IoU improvement compared with GlobalTrack at various scales. Although GlobalTrack utilizes FPN to handle scale variations while our method does not, we achieve significantly better performance on large scale objects and consistent improvement on normal-sized objects. This shows that our tracker generalizes better across scales. Our method obtains, however, slightly inferior performance at very small objects, we believe that this is due to the lack of lower-level features in our method. (b) Improvement in success score for each video at different target ratio variance ranges compared with GlobalTrack. Target ratio variance is calculated per video. Higher ratio variance shows a more drastic change in aspect ratio. VTT achieves mean success score improvements across all sequences, while for sequences with large ratio variance, VTT achieves significantly better performance than GlobalTrack.

layer has the ability to model global relations, lower layers attend to random scattered areas that are visually similar, while higher level encoder layers aggregate global information and attend to larger connected areas related to the target. This phenomenon is similar to convolution networks where lower level encoder layers focus on visual features and higher level encoder layers can aggregate global information and interpret semantics.

2) *Decoder Layers*: Since decoder is essential for converting target query to final predictions, we use at least one decoder layer in the ablation study, instead of completely disabling them like what we did for encoders. Table IV shows the result. It is worth noting that the success score (AUC) drops only 1.7 points with one decoder layer and obtains competitive results. This shows that one decoder layer can cover most simple cases. We believe that multi-layer decoders mainly benefit cases where object areas are scattered, e.g., when the target is partially occluded.

D. Discussion

1) *Handling occlusion*: For RPN-based methods, each anchor classifies whether a pre-defined region contains the target. However, during partial or heavy occlusion, the target is split into parts and they may end up in different regions, causing inaccurate predictions. In contrast, VTT benefits from global information aggregation and can model relations across long distances. Fig. 6 shows that the encoder layers can reason about the relations between disconnected regions of the same target. As illustrated in the first row of Fig. 5, even when the target is heavily occluded, VTT is able to reason the connections between distant regions of the same target and produce accurate tracking results.

2) *Generalization across scales and ratios*: GlobalTrack combines FPNs and RPNs to handle drastic changes in scale

Fig. 8. Comparison with GlobalTrack in terms of discriminative ability against instance level distractors. Higher score around ground truth and lower score around other instances denote better discriminative ability. GlobalTrack predicts higher scores for instance level distractors while ours predicts much lower scores for distractors. This shows that our method has significantly better discriminative power compared with GlobalTrack. *Best viewed in color and zoomed in.*

Fig. 9. An example video where an instance distractor is constantly present. GlobalTrack fails to discriminate between tracking target and distractor and is constantly attracted to the wrong target, while our method shows much better discriminative ability and stays on the tracking target throughout the whole sequence. *Best viewed in color.*

during tracking. However, because of the long-tailed property of object scales and ratios in training data, objects with extreme scales are rarely seen during training. As a result, GlobalTrack produces inferior results when the target undergoes significant scale or aspect ratio changes.

VTT can aggregate global information and reason about relations of different regions across long-distance, hence its performance is not affected by scales. To demonstrate this effect, we visualize decoder attention for the final prediction. As shown in the second and third rows of Fig. 5, decoder attention focuses on target extremities regardless of the target size. As a result, VTT can generalize well across different scales and aspect ratios. Fig. 7b shows that VTT achieves significant improvement over GlobalTrack in sequences with high aspect ratio variance, while Fig. 7a shows that VTT achieves consistent improvement for normal-sized targets and significant improvement for large scale objects. It does, however, obtain slightly inferior results at very small scales. We believe that this is because low-level features are not used in our method.

3) *Against instance-level distractors*: GlobalTrack utilizes FPNs to search on different scales of feature maps. This results in an astronomical amount of anchors in the RPN layer, causing extremely unbalanced samples and huge computation overhead. GlobalTrack samples negative predictions in order to address this issue. This strategy leads to loss of hard-samples. To mitigate this effect, GlobalTrack proposes a cross-query loss to increase discriminative ability against instance-level distractors. However, it requires extra annotations of other instances in the same image, making it unfeasible on tracking datasets such as the GOT-10K train set and LaSOT train set, where only one instance is annotated per video.

Our method, on the other hand, makes fewer predictions, hence it does not require any sampling. All hard cases naturally end up in the final predictions. It does not require extra human labor to annotate potential hard samples. This makes our model more discriminative against instance-level distractors. In Fig. 8, we present multiple examples to show that our tracker is significantly more discriminative compared with GlobalTrack. In Fig. 9, we present a challenging video where GlobalTrack is constantly drawn to the distractor while our method stays on the correct target throughout the whole video. This proves that our architecture is more data-efficient and more suitable for tracking tasks.

V. CONCLUSION

In this paper, we propose a new architecture with transformers for long-term visual object tracking, namely Visual Tracking Transformer (VTT). It can aggregate global information and benefit challenging tracking scenarios like occlusion and scale or ratio variation. It also shows strong discriminative ability against instance-level distractors without extra labelling or hard-sample mining. It achieves state-of-the-art performance on three large-scale long-term tracking benchmarks. We believe, as the first work to successfully address challenging long-term tracking scenarios using transformers, this new approach will benefit future research.

REFERENCES

- [1] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.
- [2] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti, "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 38–51, 2005.
- [3] S. Li and D.-Y. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *AAAI*, 2017.
- [4] M. Nashman and H. Schneiderman, "Real-time visual processing for autonomous driving," in *IV*, 1993.
- [5] B. Yan, H. Zhao, D. Wang, H. Lu, and X. Yang, "'skimming-perusal' tracking: A framework for real-time and robust long-term tracking," in *ICCV*, 2019.
- [6] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *ECCV*, 2018.
- [7] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *CVPR*, 2016.
- [8] L. Huang, X. Zhao, and K. Huang, "Globaltrack: A simple and strong baseline for long-term tracking," *arXiv preprint arXiv:1912.08531*, 2019.
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.
- [10] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," *arXiv preprint arXiv:1802.05751*, 2018.
- [11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *arXiv preprint arXiv:2005.12872*, 2020.
- [12] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *CVPR*, 2019.
- [13] A. Moudgil and V. Gandhi, "Long-term visual object tracking benchmark," in *ACCV*, 2018.
- [14] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. W. Smeulders, P. H. Torr, and E. Gavves, "Long-term tracking in the wild: A benchmark," in *ECCV*, 2018.
- [15] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *CVPR*, 2017.
- [16] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *CVPR*, 2019.
- [17] W. Huang, J. J. Gu, X. Ma, and Y. Li, "Self-paced model learning for robust visual tracking," *Journal of Electronic Imaging*, vol. 26, no. 1, pp. 013–016, 2017.
- [18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.
- [19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.
- [20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.
- [21] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *CVPR*, 2019.
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [23] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "Rwth asr systems for librispeech: Hybrid vs attention-w/o data augmentation," *arXiv preprint arXiv:1905.03072*, 2019.
- [24] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *ICCV*, 2019.
- [25] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *CVPR*, 2019.
- [26] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *CVPR*, 2015.
- [27] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *CVPR*, 2017.
- [28] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016.
- [29] Y. Zhang, D. Wang, L. Wang, J. Qi, and H. Lu, "Learning regression and verification networks for long-term visual tracking," *arXiv preprint arXiv:1809.04320*, 2018.
- [30] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *CVPR*, 2018.
- [31] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *CVPR*, 2016.
- [32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.
- [34] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, Early Access.
- [35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2018.
- [36] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. Lau, and M.-H. Yang, "Vital: Visual tracking via adversarial learning," in *CVPR*, 2018.
- [37] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV*, 2016.
- [38] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *ICCV*, 2017.