VideoTrack: Learning to Track Objects via Video Transformer -Supplementary Materials-

In this supplementary material, we first provide some implementation details mentioned in main text in Sec. 1, *e.g.* training and testing details. In Sec. 2, we then report additional VOT benchmark results including OTB [22] and VOT [12, 13] benchmarks. We also provide more exploration studies of our method through experimental and theoretical analysis and more extensive conclusions on VideoTrack framework in Sec. 3. Finally, we provide some visualization and attribute analysis of VideoTrack on tracking benchmarks and selected video sequences in Sec. 4.

1. Experiment Details

Training. We build our VideoTrack model on top of the vanilla ViT-Base [5] model pre-trained with MAE [8]. We set the video frame resolution as follows: The first template and intermediate templates all have 128×128 pixels; The search frame has 256×256 pixels. The search area factor of template and search frame are set to be 2 and 4, respectively. Then, we finetune the whole model on the tracking datasets. In particular, for each tempaltes&search frame sequence from the training datasets, we compute the losses based on the classification and regression outputs from the prediction head. We use standard cross-entropy loss for the classification loss: all pixels within the ground-truth box are regarded as positive samples and the rest are negative. We use GIoU [20] loss and L_1 loss for the regression loss. For the experimental settings with tubelet embedding or time window partition, we add paddings to the template frames in order to keep the same resolution with the search frame. For the full-dataset training, the training splits of COCO [15], LaSOT [6], GOT-10k [10] (1k forbidden sequences from GOT-10k training set are removed following the convention [23]) and TrackingNet [19] are used. Common data augmentations including horizontal flip and brightness jittering are used in training. We sample a sequence of video frames at random to construct the video-level inputs for VideoTrack. For hardware restriction and memory efficiency, we set each Tesla V100 GPU holds 16 video sequences (each has 4 frames), resulting in a total batch size of 128 for 8 GPUs. We train the model with AdamW [16] optimizer, set the weight decay to 10^{-4}, the initial learning rate for the backbone to 4×10^{-5} and other parameters to 4×10^{-4}, respectively. The total training epochs are set to 300 with 60k image pairs per epoch and we decrease the learning rate by a factor of 10 after 240 epochs. All the forbidden sequences defined by the VOT2019 challenge are abandoned. The training sequences in each iteration are sampled from one video sequence or constructed by a static image. On static images, we also construct an video sequence by applying data augmentation like flip, brightness jittering and target center jittering.

For GOT-10k [10], we only the training split of GOT-10k to train VideoTrack, following the official test protocol that no additional training datasets are allowed. It has the zero overlap of object classes between training and testing subset. The total number of epoch for training is set to 100 and the learning rate decays by a factor of 10 after 80 epochs. Other technical details are the same with the full-dataset training.

Head and loss. We adopt the similar prediction head and loss in [24, 27]. The encoded search feature is fed into a fully convolutional network (FCN), which consists of L stacked Conv-BN-ReLU layers for each output. After obtaining the fully fused feature tokens f_{zx}, we drop the template image tokens f_z and reshape the search image tokens f_x into a 2D feature map. The reshaped 2D feature map is fed into the classification and regression branches of the prediction network for corresponding outputs. Specifically, for classification, location (x_l, y_l) on feature map f_x^{cls} is considered as a positive sample if its corresponding location on the input image falls into the ground-truth bounding box. Otherwise, it is a negative sample. The total stride of the backbone is set to 16. For the regression target of each positive location (x_i, y_i) on feature map f_x^{reg}, the final layer predicts the distances from the corresponding location to the center location $(x_i - x_{gt}, y_i - y_{gt})$, height and width of the bounding box (w_i, h_i), denoted as a 4D vector $\boldsymbol{t}^* = (x^*, y^*, w^*, h^*)$. Hence, the regression targets for location (x_l, y_l) can be formulated as

$$\begin{aligned} x^* &= \mathbf{n}\{x_i + x_{gt}\}, \quad y^* = \mathbf{n}\{y_i + y_{gt}\} \\ w^* &= \mathbf{n}\{w_i\}, \quad h^* = \mathbf{n}\{h_i\} \end{aligned} \tag{1}$$

where $\mathbf{n}\{...\}$ refers to normalizing the absolute distance into

Table 1. Comparisons on additional three tracking benchmarks: OTB100 [22], VOT2018 [13] and VOT2020 [11].

Method	Year	OTB100 [18]		VOT2018 [6]			VOT2020 [19]		
		AUC	P	EAO	Acc.	Rob.	EAO	Acc.	Rob.
SiamFC [1]	2016	58.7	77.2	0.188	0.50	0.59	0.179	0.418	0.502
ATOM [4]	2019	67.1	87.9	0.401	0.59	0.20	0.271	0.462	0.734
DiMP [2]	2019	68.4	89.9	0.440	0.60	0.15	0.274	0.457	0.740
SiamRPN++ [14]	2019	69.6	91.4	0.415	0.60	0.23	-	-	-
D3S [17]	2020	-	-	0.489	0.64	0.15	0.439	0.699	0.769
Ocean [26]	2020	67.2	90.6	0.489	0.592	0.117	0.430	0.693	0.754
TrDiMP [21]	2021	-	-	0.397	0.598	0.231	-	-	-
TransT [3]	2021	69.4	-	0.302	0.59	0.42	-	-	-
Stark [3]	2021	-	-	-	-	-	0.308	0.481	0.775
SBT [7]	2022	70.9	91.5	-	-	-	0.515	0.825	0.752
VideoTrack	Ours	69.8	90.1	0.341	0.60	0.34	0.321	0.461	0.776

$[0,1]$. (x_i, y_i) and (w_i, h_i) denote the center location and box size of the predicted bounding box B^* associated with point (x_l, y_l). During training, we adopt the weighted focal loss [27] for the classification head, the ℓ_1 loss, and generalized IoU loss [20] for the regression head. Finally, the overall loss function is as follows:

$$L_{track} = L_{cls} + \lambda_{iou} L_{iou} + \lambda_{L_1} L_1, \tag{2}$$

where $\lambda_{iou} = 2$ and $\lambda_{L_1} = 5$ are the weight coefficients as in [23, 24].

Inference details. For VideoTrack, during inference, the regression head and classification head generate three response maps which embed estimated size shapes (size and offset maps) and location confidence values (classification map), respectively. The maximum confidence value and its bounding box size are chosen to be final predicted target. The templates and search image size are chosen to 128×128 and 256×256, respectively. To validate the effectiveness of VideoTrack, no other tricks such as template updating controlled by confidence value and online module are adopted.

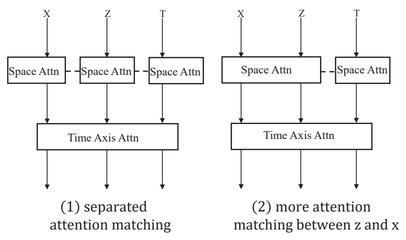
We adopt a simple online strategy in this work to avoid tricky hyper-parameters. Though our VideoTrack model is independent of the number of frames, for the current search frame F_t, we select fixed number ($T = 4$) of frames from historical frames (*i.e.* frame $F_{t-N\times T}$ to frame F_{t-1}) as intermediate templates. Each intermediate template is sampled from the nearest frame with the fixed time interval N, where we set to 30 frames here. When current time index t is smaller or over the capacity of intermediate template queue, we duplicate the first template or drop the farthest intermediate template at once.

Detailed descriptions of VOT benchmarks in main text. GOT-10k [10] is a recently released large-scale generic object tracking benchmark, containing 10,000 videos totally, in which the testing set has 180 videos. The ground truths of the testing set are also withheld so that all tracking results must be evaluated in a specific evaluation server. Different from others, GOT10k benchmark restricts trackers to use only the training set for training. GOT-10k has the zero overlap of object classes between training and testing subset. LaSOT [6] is also a large-scale single object tracking dataset with high-quality annotations. Its testing set consists of 280 long videos, with an average of 2500 frames per video. Thus, the robustness of trackers is crucial against complicated scenarios, such as occlusions, out-of-view, etc. The ranking metrics are the AUC, Precision, and Normalized Precision (P_{Norm}). TrackingNet [19] is a recent large-scale tracking benchmark consisting of 511 sequences for testing. The evaluation is performed on the online server. The testing set contains 511 videos without publicly released ground truths. We evaluate our methods on the test set of TrackingNet, which consists of 511 videos UAV123 [18] is designed to evaluate trackers in UAV applications, including 123 low altitude aerial videos, with an average of 915 frames per video. Due to the characteristics of UAV, this dataset has numerous scenarios with partial and full occlusions, out-of-view, and small objects. Thus, many objects have quite low resolutions.

2. Additional VOT Benchmark Results

In Tab. 1, to extensively evaluate the performance of VideoTrack, we further test it in additional three popular VOT benchmarks: OTB100 [22], VOT2018 [13] and VOT2020 [11].

OTB-2015 [22]. OTB-2015 is a classical benchmark in visual object tracking, containing 100 short-term videos with 590 frames per video on average. We report the results on OTB-2015. The OTB2015 dataset is known to be highly saturated over recent years As shown in Tab. 1, it is relatively harder to obtain a significant performance gain on this benchmark. The reason why the older trackers scores higher in AUC and precision is that the dataset is easy to be overfitting and contains too few testing videos. However, VideoTrack still outperforms recent strong pair-wise



case	AO I
Separated	60.2
More matching	**68.3**

Figure 1. Ablations the attention matching between z and x in building unit. AO is evaluated in GOT-10k [10].

Algorithm 1 Space-time position embedding

```
Input: video-clip input $\{F_s^i\}_{i=1}^N$, 1D learnable position embedding $P_s$
Output: Posion encoded spatotemporal tokens $\{SPT_s^i\}_{i=1}^N$
1: Patch embedding to generate tokens. $\{T_s^i\}_{i=1}^N = \mathbf{PatchEmbed}(\{F_s^i\}_{i=1}^N)$
2: Spatial encoding. $\{ST_s^i\}_{i=1}^N = \{T_s^i\}_{i=1}^N + P_s$
3: Temporal interpolation on 1D learnable position embedding. $\{P_s^i\}_{i=1}^N = \mathbf{F.interpolate}(P_s, \mathbf{size} = (T, H, W), \mathbf{mode} = '\mathbf{nearest}')$
4: Temporal encoding. $\{SPT_s^i\}_{i=1}^N = \{ST_s^i\}_{i=1}^N + P_s^i$
```

transformer-based tracker TransT which shows the effectiveness of temporal information.

VOT2018 [13]. The 2018 version of the visual object tracking (VOT) challenge contains 60 videos. It has a re-initialization testing protocol which demands the tracker to restart the the tracked target is lost. Following the evaluation protocol of the VOT2018 dataset, we report the results of our tracker in terms of expected average overlap (EAO), accuracy (A), and robustness (R) and compare it with state-of-the-art trackers. Compared with these recently proposed approaches, our VideoTrack approach still exhibits satisfactory results comparing to the transformer-based tracker TransT, which is published last year. Since the dataset is relatively small and easy to be overfit. VideoTrack does not score as much as the top-rank trackers.

VOT2020 [13]. The 2020 version of the visual object tracking (VOT) challenge contains 60 videos. It replaces the re-initialization testing protocol with one pass evaluation. The difference is that the 60 testing videos are split into multiple testing sequences to extensively test the performance of trackers. Following the evaluation protocol of the VOT2020 dataset, we report the results of our tracker in terms of expected average overlap (EAO), accuracy (A), and robustness (R) and compare it with state-of-the-art trackers. Please note that we adopt the bounding-box format instead of mask, so that the accuracy (A) score is low. For fair comparison, we compare VideoTrack to the recent state-of-the-art transformer-based tracker Stark. It shows a improvement of 1.6 points in EAO which demonstrate the strong performance of VideoTrack.

3. Additional Theoretical & Experimental Analysis on VideoTrack.

In this section, we present more theoretical and experimental Analysis on VideoTrack, including model architecture variants and video attention schemes. We hope the comprehensive analysis may inspire followers to solve VOT task from the perspective of video-level modelling and develop stronger VideoTrack models.

Why to have a hierarchical structure for the building unit ? The original ViTtrack built on vanilla ViT [5] can be directly expand to spatiotemporal domain by adding more frame-wise feature tokens into the transformer layers. This model variant can also be regarded as stacking total L joint space-time attention layers sequentially ($L = 12$ for the ViT-base [5]). The main drawbacks of this straightforward model variant lie in two aspects: One issue is that it has quadratic complexity in both space and time, *i.e.*, $O(S^2T^2)$ and the total number of layers is the maximum value (12 layers). Thus, the computation Flops rises dramatically. The another issue is that the 12 times joint space-time matching increases the burden for the model to differentiate the temporal differences among all the spatiotemporal tokens. The phenomenon that this model variant cannot benefit from more templates but suffer from the computation costs validates that the temporal contexts are not well-exploited. To solve these two issues, we have two transformer layers to formulate a basic building unit: the first layer is to conduct spatial attention within each frame and the second layer is for joint space-time matching. Thus, the computation complexity is alleviated for the quadratic complexity in both space and time is reduced to 6 transformer layers. Moreover, the separated spatial attention matching within each frame can help model to recognize the temporal differences to some extent. Based on above observations, we bring hierarchical structure to the building unit in video transformer tracking model and we further increase it to three transformer layers.

Weight-sharing video attention block or performing separated attention computation in each layer? We adopt separated attention computation in each layer for Pattern V, which has totally different hierarchical structure comparing to other model patterns (weight-sharing attention among different branches). The key difference is that the attention matching between search feature tokens x and the first template feature tokens z is conducted in all three layers of the triplet-block. As shown in Fig 1, we explore the attention matching between z and x in the building unit with two-level structure, where the first layer is to encode the spatial features and the second layer is to modelling the temporal dependencies. We find that the more atten-

case	N_T =1	N_T =2	N_T =3
12 joint space-time attention layers	**70.1**	67.1↓	66.5↓

Table 2. Ablations on joint space-time attention. N_T is the number of templates. The performance is AO in GOT-10k [10].

case (memory queue)	T=10	T=20	T=30	T=40	T=50	T=60
Pattern I	72.1	72.3	**72.7**	72.4	71.8	71.7

Table 3. Ablations on the updating interval (T temporal frames) in memory queue during inference. The performance is AO in GOT-10k [10].

tion matching between x and z can contribute to a better tracking performance as the VOT needs a strong and reliable appearance information (68.3% vs. 60.2 %). It also indicates that the improvement of temporal contexts are built on the appearance matching between template and search frame. In the case that all layers in triplet-block perform the attention matching between x and z, the dynamic information provided by the intermediate templates can further raise the tracking performance, leading to the best performance. This is also the reason why the proposed disentangled dual-template mechanism has three-layer attention matching between z and x in pattern VI.

Additional details for position embedding. The space-only position embedding follows the settings with 1D learnable position embeddings in ViT [5]. For the space-time position embedding, we expand the 1D learnable position embeddings into temporal domain through nearest neighbor interpolation. The pseudo code is shown in Alg. 1. space-time position encoding is implemented by two steps: one is the spatial position encoding for each frame-wise tokens while the second step is to encode the temporal positional differences into inter-frame tokens.

Analysis on joint space-time video attention. Since VideoTrack is build on the ViT [5] which has 12 transformer layers (L), the main drawback of this joint space-time methods is the quadratic complexity with space (S) and time (T) dimension, which is $O(LS^2T^2)$. As shown in Tab. 2, we find that the straightforward joint space-time attention cannot handle the long temporal extent which the performance goes down when more templates are used (70.1% vs. 67.1 % vs. 66.5 %). This is caused by the reason that dense space-time matching that treats every spatiotemporal token equally, is more easily influenced by the redundancy in video frames. The position encoding cannot provide enough temporal clues for complex temporal reasoning. Thus, the performance of the case with more templates is even worse than the pair-wise matching, which does not need to handle temporal contexts.

Analysis on message token video attention. Message token communication can effectively reduce the computation burden of video models. let the m denotes the pre-defined number of tokens, the computation complexity of message token communication is $O((S+M)^2(\frac{L}{2})) = O(S^2M^2L)$ (two-level structure building unit). Therefore the quadratic complexity with with space (S) and time (T) dimension is reduced to space only, which greatly ease the computation burden. However, the main drawback of message token communication is that the thorough appearance clues and context information cannot be conveyed, which is validated in the main text. Moreover, the temporal redundancy and the useful information cannot be effectively distinguished by the model, hindering it from efficient temporal message passing from templates to search frame. Here, we deeply analysis the reason why message token cannot take effects in visual tracking as it in high-level video understanding tasks. VOT task needs to exploit more static/dynamic appearance clues which is illustrated in the main text in details. The pre-defined tokens summarize the overall scene contexts from each template while it cannot decouple the information into static and dynamic clues effectively. This is also because the pre-defined tokens lose the spatial details of each frame and the attention operation are weak at local modelling that is crucial to pass spatial information.

Analysis on disentangled dual-template scheme. We analyze the computation complexity of the disentangled dual-template scheme in VideoTrack. The attention matching is conducted through all layers which results in the $O(S^2L)$ complexity. The cross/self attention among the intermediate templates results in the complexity $O(S^2T\frac{L}{3} + S^2T\frac{L}{3})$. Then the attention matching between search frame feature, static and dynamic template is $O(S^2)\frac{L}{3}$ The overall computation complexity is $O(S^2T\frac{2L}{3}+S^2\frac{L}{3}) = O(S^2T)$. Though the complexity still rises as the number of frame increasing, the disentangled dual-template scheme gets rid of quadratic complexity along with temporal dimension, *i.e.* $O(S^2T^2)$. Therefore, our mechanism can convey meaningful appearance clues thoroughly through direct cross attention matching among templates, while still keep an affordable computation costs.

Online hyper-parameters. Comparing to the labour-intensive hyper-parameters in DCF [2,4,9] methods and online template updating, our introduce less hyper-parameters to VideoTrack model. We ablate the different inference settings in adopted memory queue mechanism in Tab. 3. Though best time interval hyper-parameter can be varied in different video scenarios, we can see that VideoTrack is not too sensitive to the time interval hyper-parameter. This is in contrast to the labour-intensive and tedious hyper-parameters in temporal modelling methods [21,25] for pair-wise Siamese tracking pipeline and online optimization in DCF [2,4,9], which clearly show the strength of VideoTrack to exploit the temporal information in video frames.

The potential to be applied in other vision tasks. Template-matching mechanism has been widely applied in

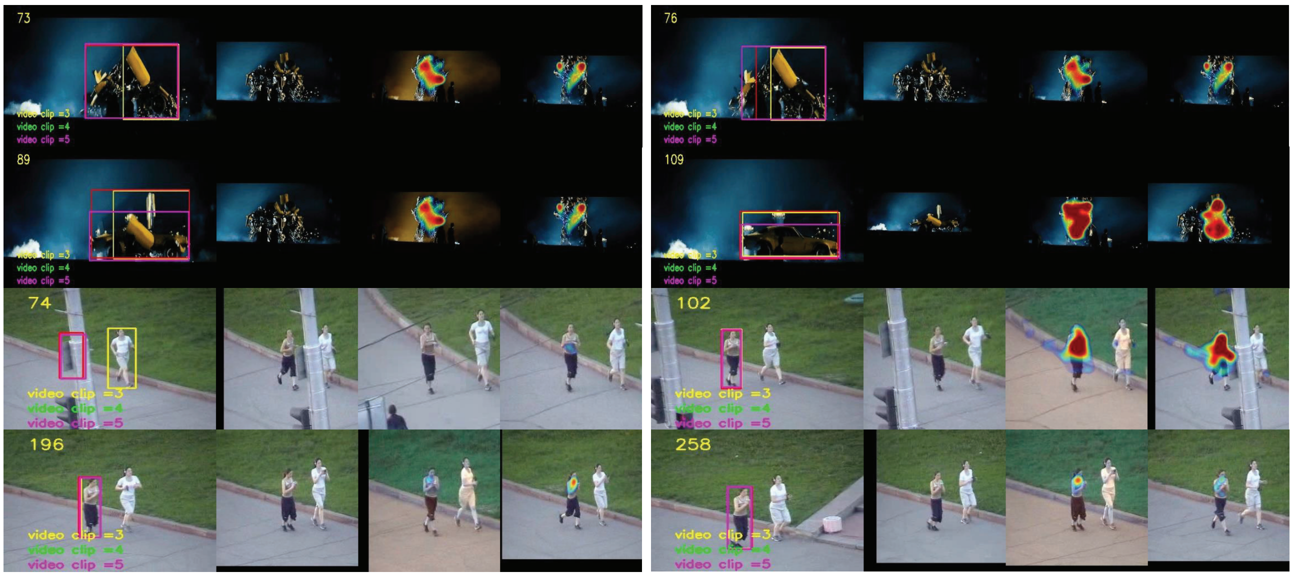


Figure 2. Visualization results of the three VideoTrack settings: the lengths of input sequence are set to 3, 4 and 5, respectively. The frame in second column is the nearest intermediate template t_T whose the center point performs as the query for the cross attention. The third and fourth row are another two intermediate templates t_{T-1} and t_1, respectively.

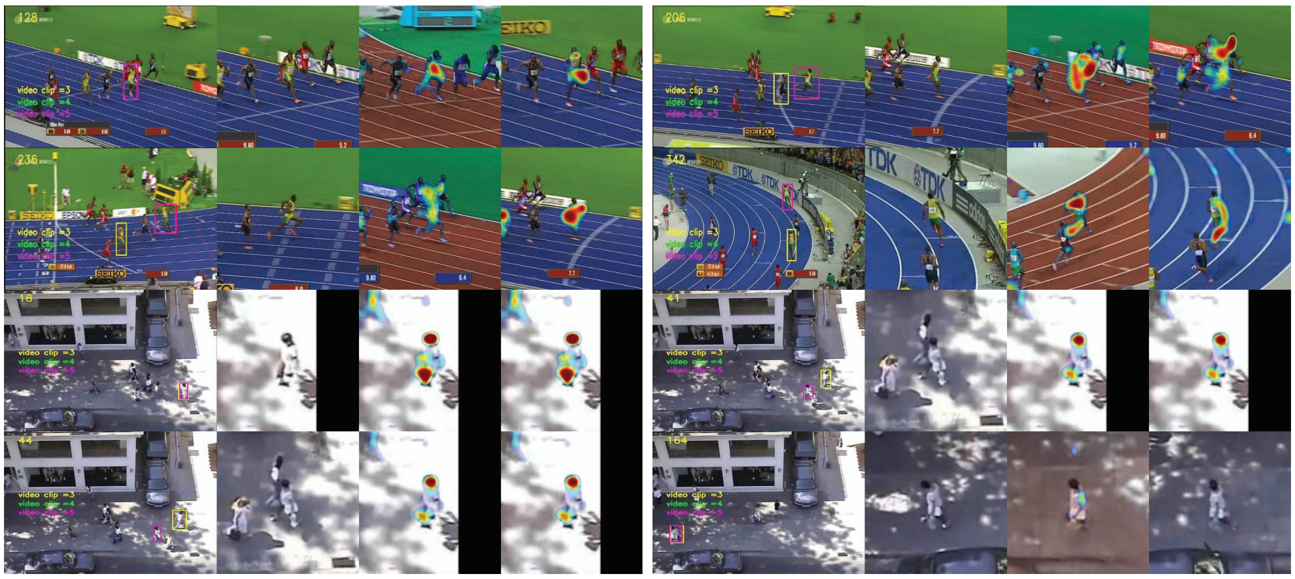


Figure 3. More visualization results of the three VideoTrack settings. Other annotations can be referred to Fig. 2.

a variety of video understanding tasks. Our VideoTrack framework can be thought as multi-template matching by a neat feedforward model. Since the VideoTrack is built with the attention block which is flexible to modelling the inter-/intra-frame dependencies, it can further be applied to other video tasks that can be improved by involving more templates, such as video segmentation and video matting. Since many works has focus on the single template matching. we will do more future works to apply VideoTrack to expand these single template-matching models.

4. Visualization and Attribute Analysis

Visualization results. As shown in Fig. 2 and Fig. 3, we visualize the VideoTrack settings with short, medium and long video sequences as inputs. The results clearly show

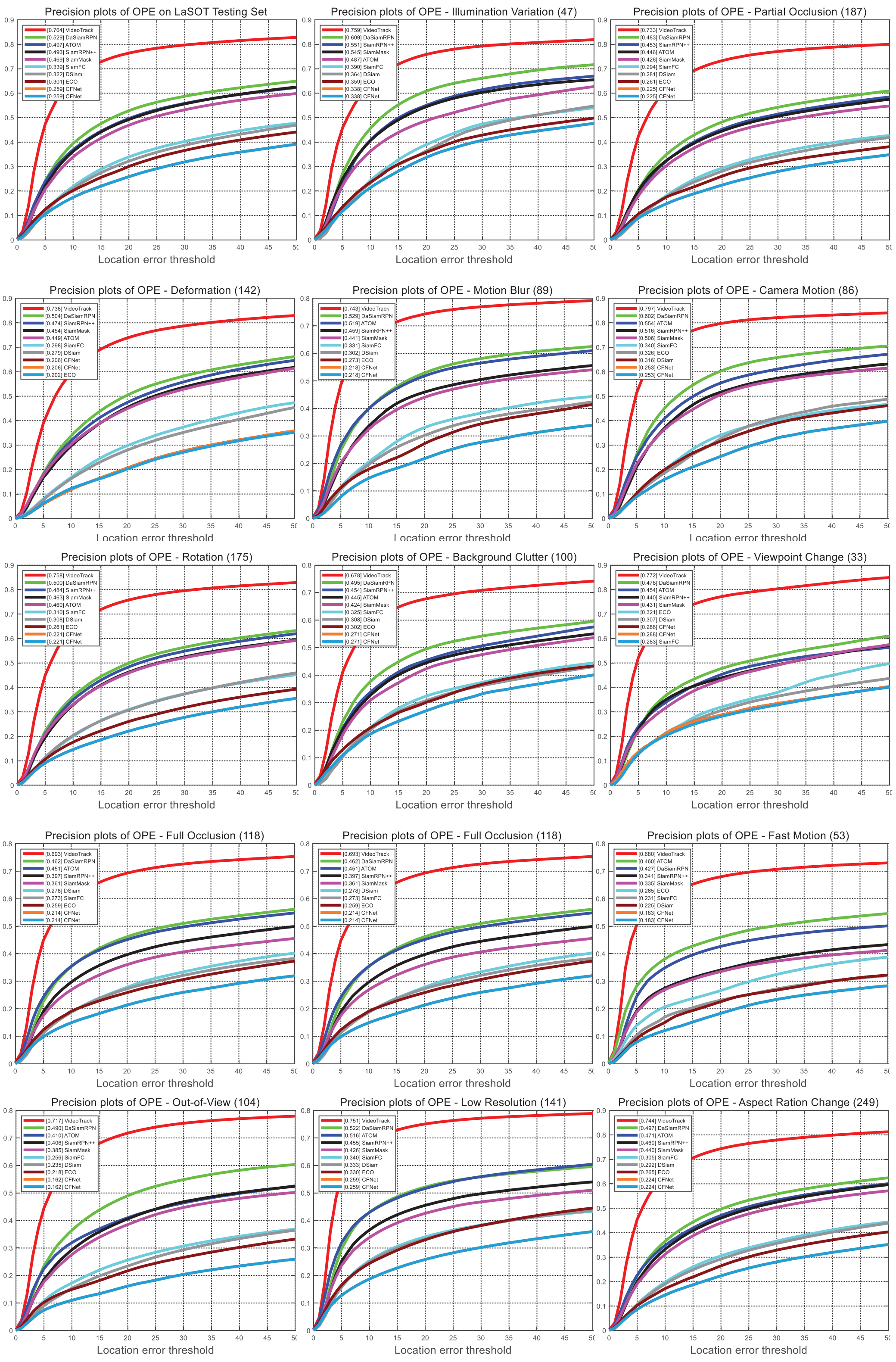


Figure 4. Attribute-based evaluation on the LaSOT benchmark [13]. The legend shows the precision scores of the precision plots.

that longer video sequences can help tracker to handle the challenging scenarios, such as appearance variations, occlusion and similar distractor objects. The most obvious phenomenon is that VideoTrack can have a more accurate size estimation of the tracked target, benefiting from the stored historical appearance information. Tracking failures are more frequent when lack of necessary temporal information. The attention activation map also shows that different video frames can improve the robustness of VideoTrack model by selectively activating feature tokens in different spatiotemporal location. Moreover, the over-long temporal extends do not provide more useful clues but redundancy, as the settings in short and long video sequence input have similar performance when the video is short or the appearance variation is not severe.

Attribute analysis. As shown in Fig. 4, we provide the attribute evaluation on the LaSOT [6] benchmark. On the LaSOT, our approaches show good results in various scenarios such as motion blur, background clutter, low resolution, and viewpoint change. It should be noted that our simple VideoTrack does not adopt complex online model optimization techniques or temporal modelling modules, which is more efficient and neat than the recent approaches such as TrDiMP [21] and TrSiam [21].

References

[1] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H S Torr. Fully-convolutional siamese networks for object tracking. In *ECCVW*, 2016. 2

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 2, 4

[3] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021. 2

[4] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019. 2, 4

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 3, 4

[6] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019. 1, 2, 7

[7] Xie Fei, Wang Chunyu, Wang Guangting, Cao Yue, Yang Wankou, and Zeng Wenjun. Correlation-aware deep tracking. In *CVPR*, 2022. 2

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1

[9] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. In *ICVS*, 2008. 4

[10] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 43(5):1562–1577, 2019. 1, 2, 3, 4

[11] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, Ondrej Drbohlav, et al. The eighth visual object tracking vot2020 challenge results. In *ECCV*. Springer, 2020. 2

[12] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Martin Danelljan, Alan Lukezic, Ondrej Drbohlav, Linbo He, Yushan Zhang, Song Yan, Jinyu Yang, Gustavo Fernandez, and et al. The eighth visual object tracking vot2020 challenge results, 2020. 1

[13] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka ˇCehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *ICCV*, 2018. 1, 2, 3

[14] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019. 2

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[17] Alan Lukezic, Jiri Matas, and Matej Kristan. D3S-a discriminative single shot segmentation tracker. In *CVPR*, 2020. 2

[18] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016. 2

[19] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018. 1, 2

[20] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 1, 2

[21] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, 2021. 2, 4, 7

[22] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 1, 2

[23] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, 2021. 1, 2

[24] Botao Ye, Hong Chang, Bingpeng Ma, and Shiguang Shan. Joint feature learning and relation modeling for tracking: A one-stream framework. *arXiv preprint arXiv:2203.11991*, 2022. 1, 2

[25] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, 2019. 4

[26] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 2

[27] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 2