Supplementary Materials for Global Tracking via Ensemble of Local Trackers

Zikun Zhou[1,*], Jianqiu Chen[1,*], Wenjie Pei[1,†], Kaige Mao[1], Hongpeng Wang[1,2], and Zhenyu He[1,†]
[1]Harbin Institute of Technology, Shenzhen [2]Peng Cheng Laboratory

In the supplementary materials, we first detail the implementation of the deformable attention-based local tracker. Then, we describe the training and inference details of our algorithm. Finally, we provide more experimental results and more discussions.

1. Implementation of the Local Trackers

The parallel local trackers in our global tracking framework are implemented with a transformer decoder based on the multi-head self-attention [11] and the deformable cross-attention [18] operations. Figure 1 shows the implementation of the parallel local trackers (transformer decoder). Similar to [1, 18], we stack $M = 6$ identical layers of the self-attention and the deformable cross-attention to build the decoder. Note that we only formulate one single layer of them in Eq. (3) in the main paper for presentation clarity. In this implementation, the default target queries are vectorial embeddings offline learned from large-scale datasets. The default reference positions are predicted by feeding the default target queries into a fully-connected layer and the sigmoid function. Searching for the target in the local region around the reference position is achieved by the deformable cross-attention [18] operation. Herein the deformable cross-attention performs the interaction between the target queries and the sparse feature pixels around the reference positions sampled from the encoded feature to output the target embeddings further used to predict target candidates. Particularly, the sample positions around the reference position are produced from the corresponding target query by predicting the coordinate offsets to the reference position via a fully-connected layer. For more details of the deformable cross-attention, we refer the readers to [18] for a more comprehensive understanding.

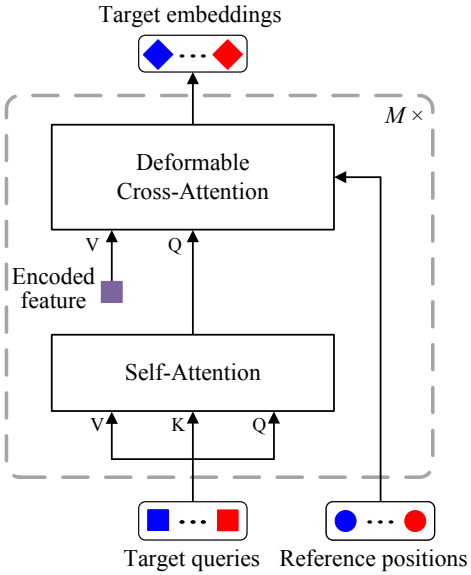

Figure 1. Illustration of the implementation of the parallel local trackers (transformer decoder). The parallel local trackers are implemented with a transformer decoder based on the self-attention and the deformable cross-attention. For clarity, we only show the core attention modules and omit the FFN blocks.

2. Training and Inference Details

Training Details. As mentioned in the main paper, we use the sequence training sample that includes one template image and one or more testing images to train our model. The length of the sequence training sample is set to 2 at the beginning of the training and then will be increased to 3, 4, 5, 6 at the 100^{th}, 175^{th}, 225^{th}, and 275^{th} epochs, respectively. The training splits of COCO [9], LaSOT [4], TrackingNet [12], and GOT-10k [6] are used to train our model, and COCO [9] is only used when the length of the sequence training sample is 2. In a sequence training sample, the template image, corresponding to 2^2 times of the target box area, is resized to 128×128, while the full testing image is resized to 640×480. Data augmentations, including brightness jittering, random translation, and random rescaling, are used. We use AdamW [10] with a weight decay of 1×10^{-4} to optimize the parameters in our model. The learning rates of the backbone parameters and the remaining parameters are 1×10^{-5} and 1×10^{-4}, respectively. The learning rate drops by a factor of 10 after 300 epochs. The number of total training epochs is 325. In each epoch, we train our model with 6×10^4 randomly sampled sequence training samples. We use 8 RTX 3090 GPUs to train our global

*Equal contribution. †Corresponding authors: Wenjie Pei and Zhenyu He (wenjiecoder@outlook.com and zhenyuhe@hit.edu.cn).

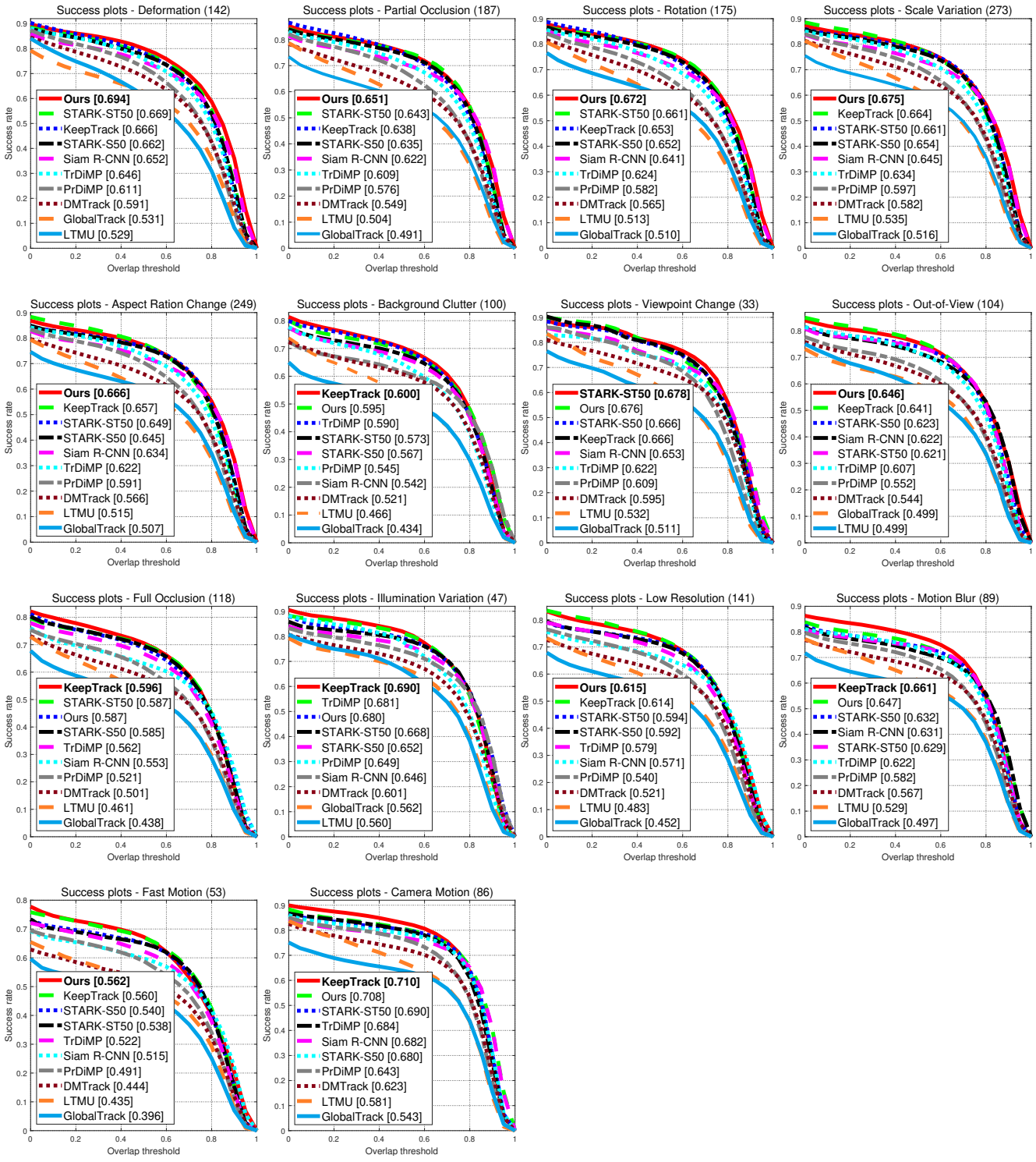

Figure 2. **Success plots over different attributes on the test set of LaSOT.** From top left to bottom right, the figures are with the challenges of deformation, partial occlusion, rotation, scale variation, aspect ratio change, background clutter, viewpoint change, out-of-view, full occlusion, illumination variation, low resolution, motion blur, fast motion, and camera motion, respectively. Our global tracking algorithm performs well on these challenging attributes.

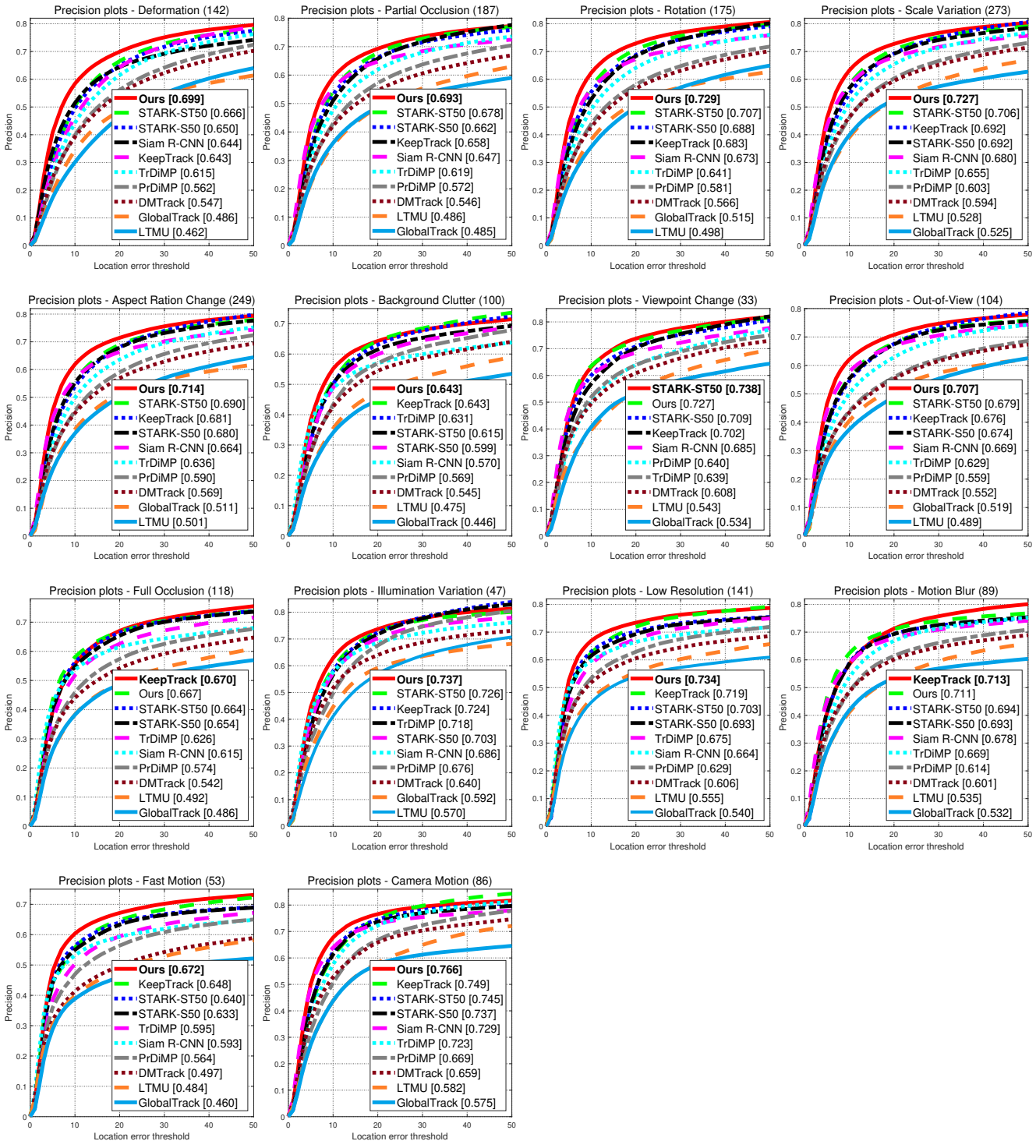

Figure 3. **Precision plots over different attributes on the test set of LaSOT.** From top left to bottom right, the figures are with the challenges of deformation, partial occlusion, rotation, scale variation, aspect ratio change, background clutter, viewpoint change, out-of-view, full occlusion, illumination variation, low resolution, motion blur, fast motion, and camera motion, respectively. Our global tracking method achieves favorable performance on these challenging attributes.

tracking model. The batch size on a single GPU is set to 14, 7, 5, 3, and 3 corresponding to different lengths of the sequence training sample. The training phase of our global tracking model takes about 110 hours.

Inference Details. During tracking, the head model in our framework outputs N (which is 10 in our setting) candidate bounding boxes and corresponding classification scores. Besides, we compute the cosine similarity between every candidate and the target template in the feature space. Specifically, we use a lightweight projection network consisting of two convolutional layers and two FC layers to convert the features of the target template and the candidates into vectorial embeddings and then calculate the cosine similarity between them. We first dump the patch features extracted by our global tracking model as the training samples, and then uses these dumped samples to train the projection network. The final confidence score of the candidate is obtained by calculating the product of the classification score and the cosine similarity. To select the final tracking result from the candidates, we consider two kinds of clues: the confidence score of the candidate and the movement between adjacent frames, similar to [17]. With these clues, we adopt a Hungarian algorithm [8] to match the target prediction in the previous frame with the candidates in the current frame. To evaluate our method on VOT2020-LT [7] which requires trackers to output a confidence score for their prediction, we directly output the confidence score of the selected candidate. To evaluate our method on OxUvA [13] which requires trackers to predict whether the target is visible, we compare the confidence score of the selected candidate with a threshold of 0.6. For the other benchmarks, we just output the selected candidate bounding box. The source codes and raw results will be available at https://github.com/ZikunZhou/GTELT.

3. More Experimental results

Herein we provide the attribute-based experimental results on the LaSOT [4] dataset. Figure 2 and Figure 3 show the attribute-based success and precision plots, respectively. The annotated attributes include deformation, partial occlusion, rotation, scale variation, aspect ratio change, background clutter, viewpoint change, out-of-view, full occlusion, illumination variation, low resolution, motion blur, fast motion, and camera motion. The trackers involved in the comparison include three global trackers (Siam R-CNN [14], DMTrack [17], and GlobalTrack [5]), one local-global switching strategy tracker (LTMU [2]), and four local trackers (STARK-ST50 [16], STARK-S50 [16], TrDiMP [15], PrDiMP [3]). Our method performs favorably against the state-of-the-art trackers on most attributes.

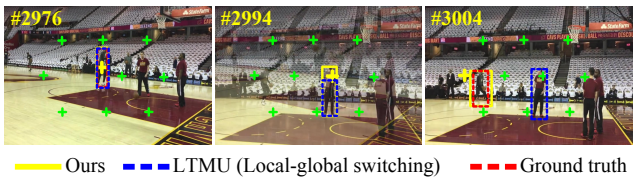

Figure 4. Qualitative comparison between our method and the local-global switching strategy method LTMU on dealing with the distractor. When the target disappears in the 2994^{th} frame due to shot cut, both our method and LTMU drift to a nearby distractor. However, when the target reappears in the 3004^{th} frame, our method successfully recovers to the target while LTMU keeps tracking the distractor.

4. More Discussions about the Distractor Issue

As pointed out in the main paper, the proposed global tracking framework cannot handle the extremely challenging distractor issue. For example, in the situation there is an existing similar distractor when the target disappears in the view, our method may drift to the distractor after losing the target. Such a complicated situation is also quite challenging for most local-global switching strategy trackers. However, when the target reappears, our method with a global view is more likely to recover to the target than the local-global switching tracker, which may still track the distractor in a local view without sensing the target. We show such a case in Figure 4.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[2] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In *CVPR*, pages 6298–6307, 2020.

[3] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, pages 7183–7192, 2020.

[4] 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*, pages 5374–5383, 2019.

[5] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Globaltrack: A simple and strong baseline for long-term tracking. In *AAAI*, volume 34, pages 11037–11044, 2020.

[6] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE TPAMI*, 43(5):1562–1577, 2021.

[7] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, Ondrej Dr-

bohlav, et al. The eighth visual object tracking vot2020 challenge results. In *ECCV*, pages 547–601. Springer, 2020.

[8] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[11] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[12] 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*, pages 300–317, 2018.

[13] Jack Valmadre, Luca Bertinetto, Joao F Henriques, Ran Tao, Andrea Vedaldi, Arnold WM Smeulders, Philip HS Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. In *ECCV*, pages 670–685, 2018.

[14] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *CVPR*, pages 6578–6588, 2020.

[15] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, pages 1571–1580, 2021.

[16] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, pages 10448–10457, 2021.

[17] Zikai Zhang, Bineng Zhong, Shengping Zhang, Zhenjun Tang, Xin Liu, and Zhaoxiang Zhang. Distractor-aware fast tracking via dynamic convolutions and mot philosophy. In *CVPR*, pages 1024–1033, 2021.

[18] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020.