Saliency-Associated Object Tracking Supplementary Materials

Zikun Zhou[1], Wenjie Pei[1,*], Xin Li[2], Hongpeng Wang[1,2], Feng Zheng[3], and Zhenyu He[1,*]
[1]Harbin Institute of Technology, Shenzhen [2]Peng Cheng Laboratory
[3]Southern University of Science and Technology
zhouzikunhit@gmail.com wenjiecoder@outlook.com xinlihitsz@gmail.com
wanghp@hit.edu.cn zfeng02@gmail.com zhenyuhe@hit.edu.cn

In this document, we first formulate the process of the back-propagation of saliency evaluation, and then provide more detailed experimental results in terms of both qualitative and quantitative evaluations.

1. Back-propagation of Saliency evaluation

During the end-to-end training of our whole model, the gradients are back-propagated to the similarity maps $\boldsymbol{S}$ through two parallel paths in the Saliency Mining module: the path through the Saliency Evaluator and the path directly backward to $\boldsymbol{S}$ in Figure 2. Here, we show the back-propagation for one channel of similarity map $\boldsymbol{S}_{(u,v)}$ for the pixel located at (u, v) in $\boldsymbol{F}_x$. We denote $\boldsymbol{S}_{(u,v)}$ as $\boldsymbol{E}$ for simplicity, and then the normalized similarity map by the saliency score $s(\boldsymbol{E})$ is $\boldsymbol{E}' = s(\boldsymbol{E}) \cdot \boldsymbol{E}$, where $s(\boldsymbol{E})$ is the saliency score obtained by our Saliency Evaluator:

$$s(\boldsymbol{E}) = \gamma(\boldsymbol{E})[c(\boldsymbol{E})]^{\alpha} + \lambda g_{\mu_g,\sigma_g}(u, v). \tag{1}$$

Thus we aim to back-propagate the gradient of $\boldsymbol{E}'$ w.r.t. $\boldsymbol{E}$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{E}'}\frac{\partial \boldsymbol{E}'}{\partial \boldsymbol{E}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{E}'}\, s(\boldsymbol{E}) + \frac{\partial \mathcal{L}}{\partial \boldsymbol{E}'}\frac{\partial s(\boldsymbol{E})}{\partial \boldsymbol{E}}\boldsymbol{E}, \tag{2}$$

where $\frac{\partial \mathcal{L}}{\partial \boldsymbol{E}'}$ is the gradient from the loss $\mathcal{L}$. In Eq. 2, the first term corresponds to the gradient in the path directly back-propagated to $\boldsymbol{E}$, while the second term corresponds to the path through the Saliency Evaluator. Since $c(\boldsymbol{E})$ in Eq. 1 is not differentiable w.r.t. $\boldsymbol{E}$ (check the calculation of $c(\boldsymbol{E})$ in the paper), we ignore the gradients of $c(\boldsymbol{E})$ w.r.t. $\boldsymbol{E}$, and then the gradients of $s(\boldsymbol{E})$ w.r.t $\boldsymbol{E}$, $\frac{\partial s(\boldsymbol{E})}{\partial \boldsymbol{E}}$, are calculated by:

$$\begin{aligned}\frac{\partial s(\boldsymbol{E})}{\partial \boldsymbol{E}} &= \frac{\partial \gamma(\boldsymbol{E})}{\partial \boldsymbol{E}}[c(\boldsymbol{E})]^{\alpha}, \\ \frac{\partial \gamma(\boldsymbol{E})}{\partial \boldsymbol{E}} &= \frac{\sigma_\Phi(\frac{\partial \max(\boldsymbol{E})}{\partial \boldsymbol{E}} - \frac{\partial \mu_\Phi}{\partial \boldsymbol{E}}) - (\max(\boldsymbol{E}) - \mu_\Phi)\frac{\partial \sigma_\Phi}{\partial \boldsymbol{E}}}{(\sigma_\Phi)^2}.\end{aligned} \tag{3}$$

Herein $\frac{\partial \max(\boldsymbol{E})}{\partial \boldsymbol{E}}$, $\frac{\partial \mu_\Phi}{\partial \boldsymbol{E}}$, and $\frac{\partial \sigma_\Phi}{\partial \boldsymbol{E}}$ can be readily obtained by the built-in Pytorch implementations. So far it has been shown how to calculate the gradient of $\boldsymbol{E}'$ w.r.t. $\boldsymbol{E}$ and how the whole model can be trained in an end-to-end manner.

*Corresponding authors.

2. More Qualitative Studies

To give more insights into our method, we visualize the correlation representations produced by our *SAOT* and four variants (DW-Corr, PG-Corr, PPFM, and PAM) described in Section 4.2 in the manuscript. Figure 1 compares the correlation features and the bounding boxes on four challenging sequences. It shows that PAM is able to generate more precise and smooth correlation features than those by PPFM, and our *SAOT* further improves the precision of the correlation features for reflecting target states and the robustness to distractors compared with PAM. These comparisons visually demonstrate the effectiveness of the proposed association modeling method and the saliency mining mechanism. Besides, the proposed *SAOT* is able to learn more precise correlation features than those computed by DW-Corr and PG-Corr. As a result, our *SAOT* predicts more precise bounding boxes than those by the other four variants.

3. More Quantitative Results

In this section, we provide more detailed quantitative results including 1) the precision and success plots on the NFS30 [9] dataset; 2) the attribute-based plots in terms of success on the OTB2015 [14] dataset; 3) the attribute-based plots in terms of success on the testing set of LaSOT [6].

3.1. NFS30

The NFS [9] dataset has a 240 FPS version (NFS240) and a 30 FPS version (NFS30). We evaluate the proposed *SAOT* on the 30 FPS version. Figure 2 shows the precision and success plots of our *SAOT* and four representative state-of-the-art trackers, including KYS [2], PrDiMP [5], DiMP [1], and SiamBAN [3]. The proposed algorithm achieves the best performance in terms of both precision and success.

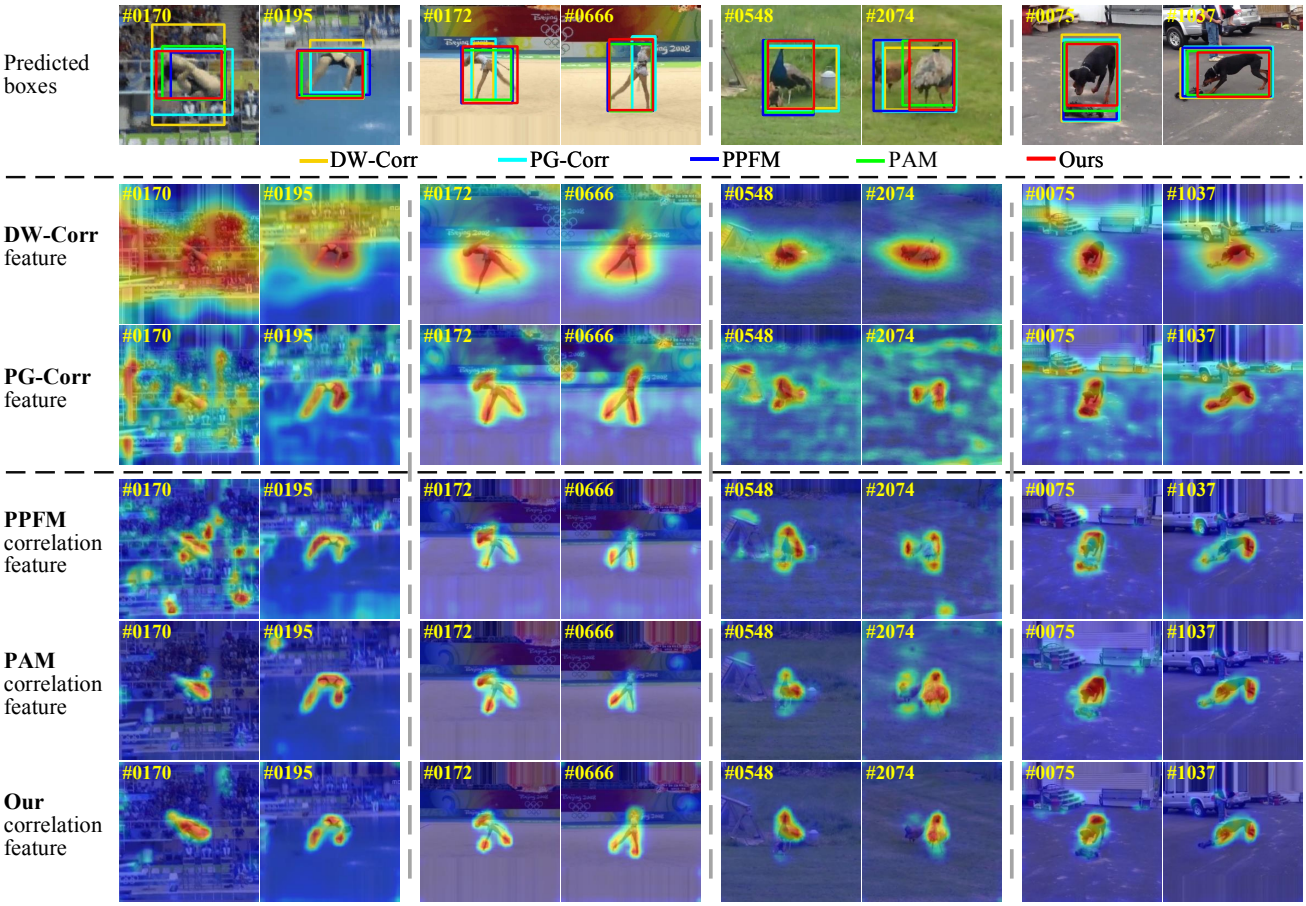

Figure 1. **Qualitative comparison between DW-Corr, PG-Corr, PPFM, PAM, and *SAOT* on four challenging tracking sequences (left two with deformation and the other two with distractors).** Our *SAOT* can learn more precise correlation features than those produced by the other four variants. Consequently, our *SAOT* predicts more precise bounding boxes than those estimated by the other four variants.

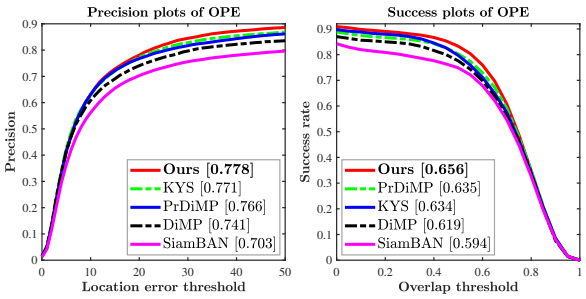

Figure 2. **Precision and success plots of different tracking algorithms on the NFS30 dataset.**

3.2. OTB2015

Figure 3 shows the attribute-based success plots on the OTB2015 dataset. The attributes include deformation, occlusion, out-plane rotation, in-plane rotation, scale variation, background clutter, fast motion, illumination variation, low resolution, out of view, and motion blur. The methods involved in the comparisons include Ocean [15], KYS [2], PrDiMP [5], DiMP [1], GradNet [12], ATOM [4], GCT [8], SiamRPN++ [10], and SiamRPN [11]. The proposed method performs favorably against the state-of-the-art tracking algorithms on these challenging attributes. In particular, our *SAOT* achieves performance gains of 1.8% and 2.8% compared with the second-best trackers (KYS and SiamRPN++) on the attributes of deformation and occlusion, respectively.

3.3. LaSOT

Figure 4 shows the attribute-based success plots on the test set of LaSOT [6]. 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 tracking algorithms involved in the comparisons include Ocean [15], KYS [2], PrDiMP [5], SiamBAN [3], DiMP [1], ATOM [4], SiamRPN++ [10], SPM [13], and C-RPN [7]. Our *SAOT* performs well on these challenging attributes.

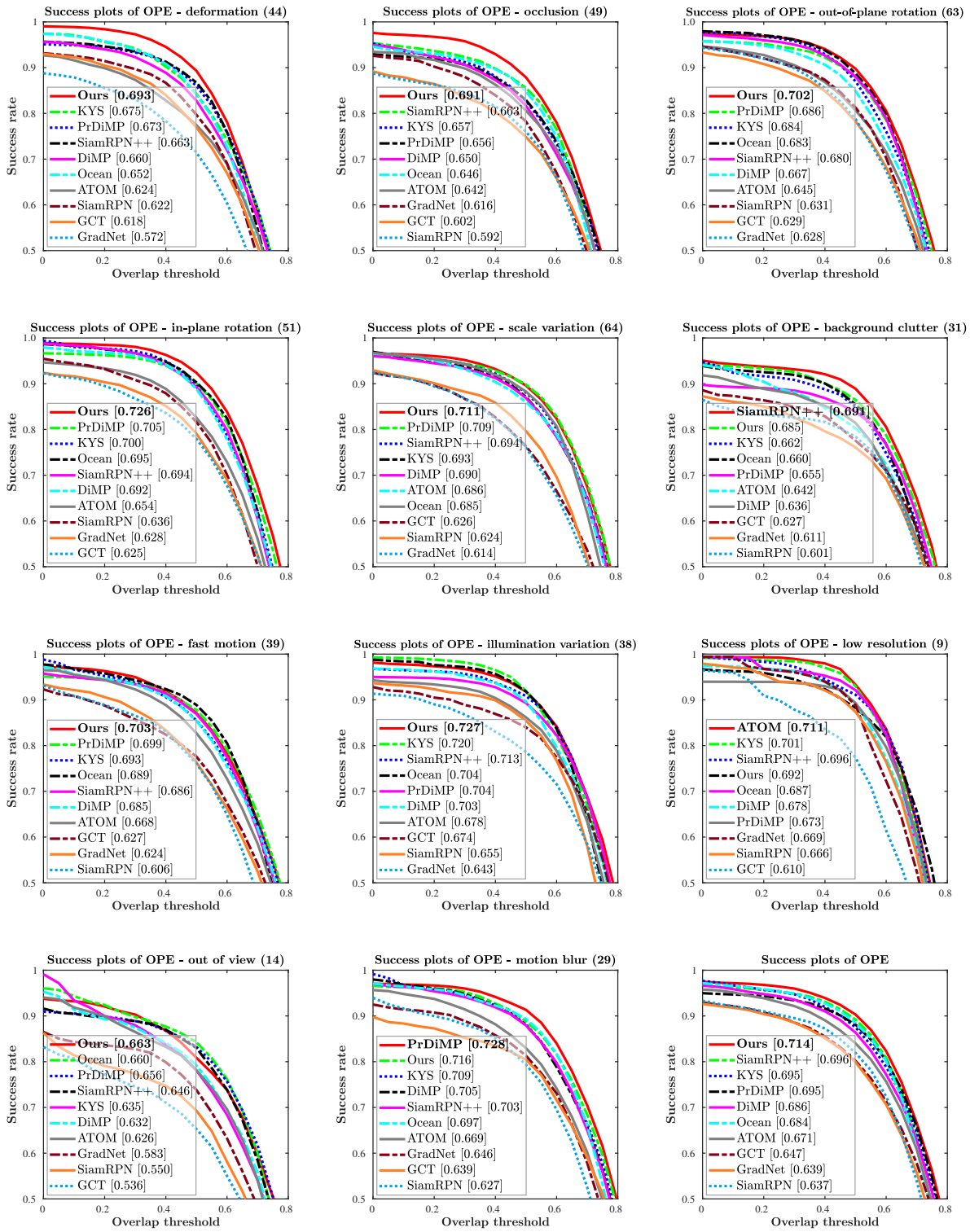

Figure 3. **Success plots over different attributes on the OTB2015 dataset.** From top left to bottom right, the figures are with the challenges of deformation, occlusion, out-plane rotation, in-plane rotation, scale variation, background clutter, fast motion, illumination variation, low resolution, out of view, motion blur, respectively. The last figure shows the overall performance. The proposed approach performs favorably against the state-of-the-art methods on these challenging attributes.

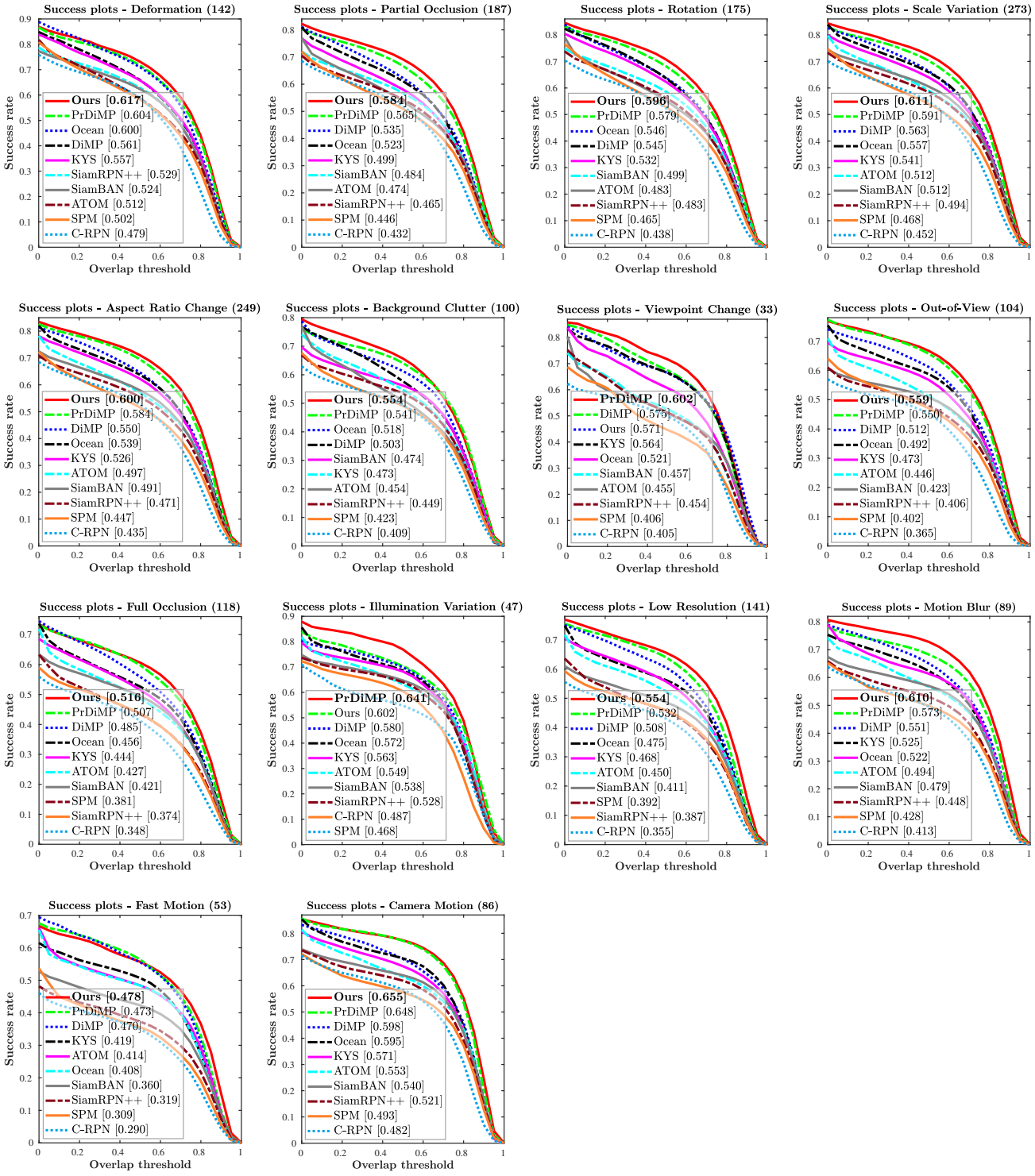

Figure 4. **Success plots over different attributes on the test set of LaSOT.** From top left to bottom right, the figures are with the challenges of deformation, partial occlusion, rotation, scale variation, aspect ratio change, background clutter, viewpoint change, out of view, full occlusion, illumination variation, low resolution, motion blur, fast motion, and camera motion, respectively. Our *SAOT* performs well on these challenging attributes.

References

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

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, 2020.

[3] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, 2020.

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

[5] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020.

[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.

[7] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019.

[8] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. Graph convolutional tracking. In *CVPR*, 2019.

[9] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, 2017.

[10] 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.

[11] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[12] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Gradnet: Gradient-guided network for visual object tracking. In *ICCV*, 2019.

[13] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Spm-tracker: Series-parallel matching for real-time visual object tracking. In *CVPR*, 2019.

[14] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

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