Supplementary Material
Progressive Unsupervised Learning for Visual Object Tracking

Qiangqiang Wu Jia Wan Antoni B. Chan
Department of Computer Science, City University of Hong Kong
qiangqwu2-c@my.cityu.edu.hk, jiawan1998@gmail.com, abchan@cityu.edu.hk

In this supplementary material, we provide additional visualization, derivations and implementation details. Section A shows the mini-batch training samples selected by the proposed anchor-based hard negative mining (AHM) strategy. Section B contains detailed derivations of 2nd-order Taylor Approximation for the expectation. We qualitatively compare the proposed AHM with the original random selection for contrastive learning in Section C. More qualitative and quantitative results are given in Section D to demonstrate the robustness of our proposed noise-robust (NR) loss to observation noise. We further show the qualitative tracking results and failure cases in Section E. Finally, we discuss the additional implementation details in Section F.

A. Mini-batch Samples Selected by AHM

Fig. 1 shows the samples (without applying data augmentation) selected by the proposed AHM. With AHM, the anchor sample (the top-left sample shown in Fig. 1) with close nearest neighbors can be selected. We use the selected anchor sample and its nearest neighbors as one mini-batch training samples for more challenging contrastive learning, which facilitates the model to identify hard negative distractors.

B. Detailed 2nd-order Taylor Approximation

In this subsection, we detail the derivations of 2nd-order Taylor approximation in Section 3.4.2. As described in the paper, by applying a 2nd-order Taylor approximation to exp, we have:

$$\log p(\mathbf{m}) \approx \log(1 + \mathbb{E}_{\boldsymbol{\epsilon}}[g(\boldsymbol{\epsilon})] + \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}}[g(\boldsymbol{\epsilon})^2]). \quad (1)$$

Note that we define the following notation in the paper:

$$\begin{aligned} \mathbf{m}_1(\mathbf{x}) &= \mathbf{m}(\mathbf{x}), \quad & \mathbf{m}_0(\mathbf{x}) &= 1 - \mathbf{m}(\mathbf{x}), \\ \mathbf{h}_1(\mathbf{x}) &= \log \mathbf{f}(\mathbf{x}), \quad & \mathbf{h}_0(\mathbf{x}) &= \log(1 - \mathbf{f}(\mathbf{x})), \end{aligned} \quad (2)$$

and thus $g(\boldsymbol{\epsilon}) = \sum_{\mathbf{x}} \sum_{b\in\{0,1\}} \mathbf{m}_b(\mathbf{x})\mathbf{h}_b(\mathbf{x})$. By substituting into the first expectation $\mathbb{E}_{\boldsymbol{\epsilon}}[g(\boldsymbol{\epsilon})]$, yielding:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\epsilon}}[g(\boldsymbol{\epsilon})] &= \mathbb{E}_{\epsilon}[\sum_{\mathbf{x}} \sum_{b\in\{0,1\}} \mathbf{m}_b(\mathbf{x})\mathbf{h}_b(\mathbf{x})] \\ &= \sum_{x} \mathbb{E}_{\epsilon}[\mathbf{m}_1(\mathbf{x})]\mathbf{h}_1(\mathbf{x}) + \mathbb{E}_{\epsilon}[\mathbf{m}_0(\mathbf{x})])\mathbf{h}_0(\mathbf{x}) \\ &= \sum_{\mathbf{x}} \sum_{b\in\{0,1\}} \bar{\mathbf{m}}_b(\mathbf{x})\mathbf{h}_b(\mathbf{x}). \end{aligned} \quad (3)$$

For the 2nd expectation,

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\epsilon}}[g(\boldsymbol{\epsilon})^2] = &= \mathbb{E}_{\epsilon}\left[\left\{\sum_{\mathbf{x}} \sum_{b\in\{0,1\}} \mathbf{m}_b(\mathbf{x})\mathbf{h}_b(\mathbf{x})\right\}^2\right] \\ &= \mathbb{E}_{\epsilon}\left[\sum_{\mathbf{x}}\sum_{\mathbf{x}'}\sum_{b}\sum_{b'} \mathbf{m}_b(\mathbf{x})\mathbf{m}_{b'}(\mathbf{x}')\mathbf{h}_b(\mathbf{x})\mathbf{h}_{b'}(\mathbf{x}')\right] \\ &= \sum_{\mathbf{x}}\sum_{\mathbf{x}'}\sum_{b}\sum_{b'} \mathbb{E}_{\epsilon}\left[\mathbf{m}_b(\mathbf{x})\mathbf{m}_{b'}(\mathbf{x}')\mathbf{h}_b(\mathbf{x})\mathbf{h}_{b'}(\mathbf{x}')\right] \\ &= \sum_{\mathbf{x},\mathbf{x}',b,b'} \mathbf{V}_{b,b'}(\mathbf{x},\mathbf{x}')\mathbf{h}_b(\mathbf{x})\mathbf{h}_{b'}(\mathbf{x}'), \end{aligned} \quad (4)$$

where $\mathbf{V}_{b,b'}$ is a 2nd-moment matrix,

$$\mathbf{V}_{b,b'}(\mathbf{x},\mathbf{x}') = \mathbb{E}_{\boldsymbol{\epsilon}}[\mathbf{m}_b(\mathbf{x})\mathbf{m}_{b'}(\mathbf{x}')]. \quad (5)$$

C. Comparison with Random Selection

We visualize the samples selected by our proposed AHM and the original random selection strategy (RSS) in Fig. 2. Our AHM tends to select very similar mini-batch samples (see Fig. 1), while RSS chooses random samples for contrastive learning, which causes the denominator in the contrastive loss (Eq. 4 in the paper) to be relatively small, especially when using a small batch size. In the original contrastive learning task [2], RSS is commonly used with

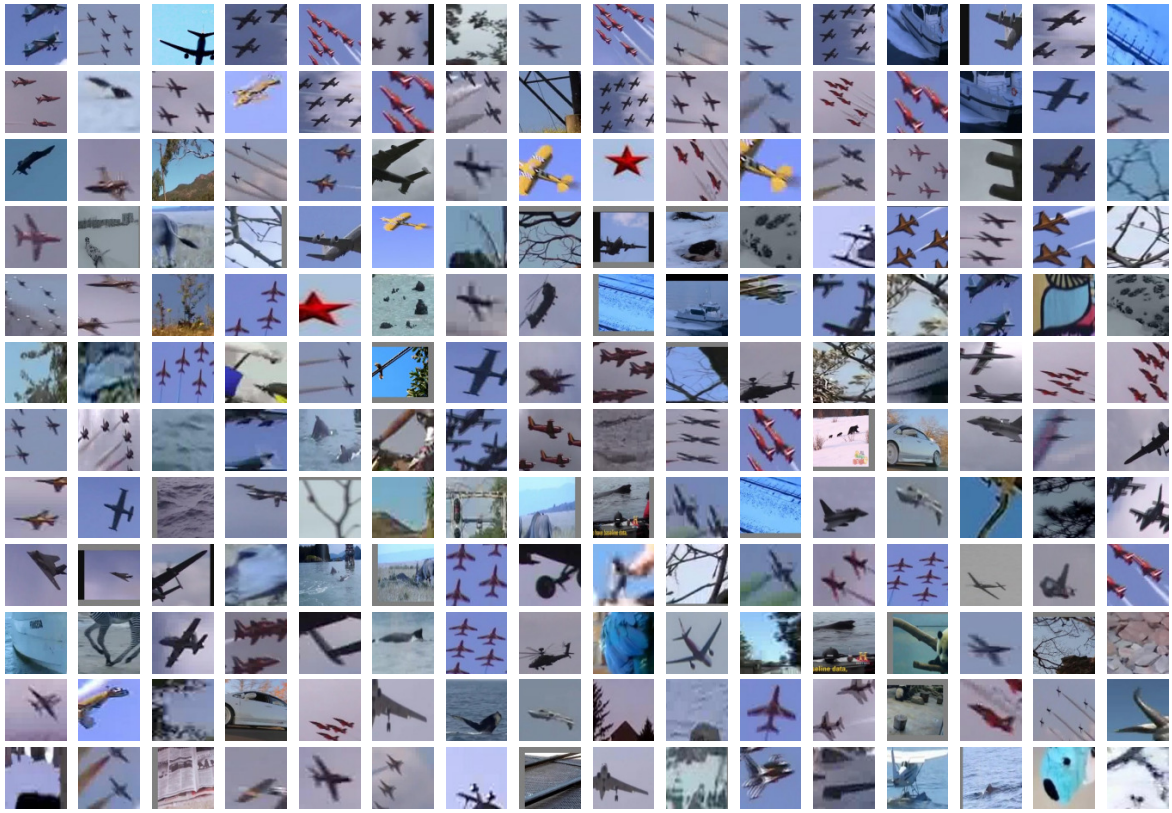

Figure 1: Example mini-batch training samples (without applying data augmentation) selected by our AHM for contrastive learning. The top-left patch is the anchor sample.

Figure 2: Partial example mini-batch training samples selected by the random selection strategy for contrastive learning.

a large batch size (e.g., 8192), which guarantees that rich negative samples are selected and the denominator in the contrastive loss is large enough for more effective learning. However, using a large training batch size is not memory-friendly. To alleviate this issue, we propose to select hard negative samples for more effective contrastive learning even with a relatively small batch size.

D. Robustness to Observation Noise

Fig. 3 shows the comparison between our proposed NR loss and the original BCE loss with different noise levels. Note that the experimental setting are kept the same as the original experiment in Section 4.2.2. In Fig. 3, the proposed NR loss can well handle different levels of observation noises, and meanwhile it performs better than the BCE loss for noisy sample learning. We further make the comparison between the two losses with more noisy samples in Fig. 4. Even with a very large noise (i.e., 50 pixels), the tracking model learned with the proposed NR loss can effectively track the objects, while the BCE loss fails to learn a robust tracking model from these noisy samples.

In Fig. 6, we report additional tracking results when training with various amounts of spatial annotation noise using traditional binary cross entropy (BCE) loss and our noise-robust loss (as in Section 4.2.2). When no noise is

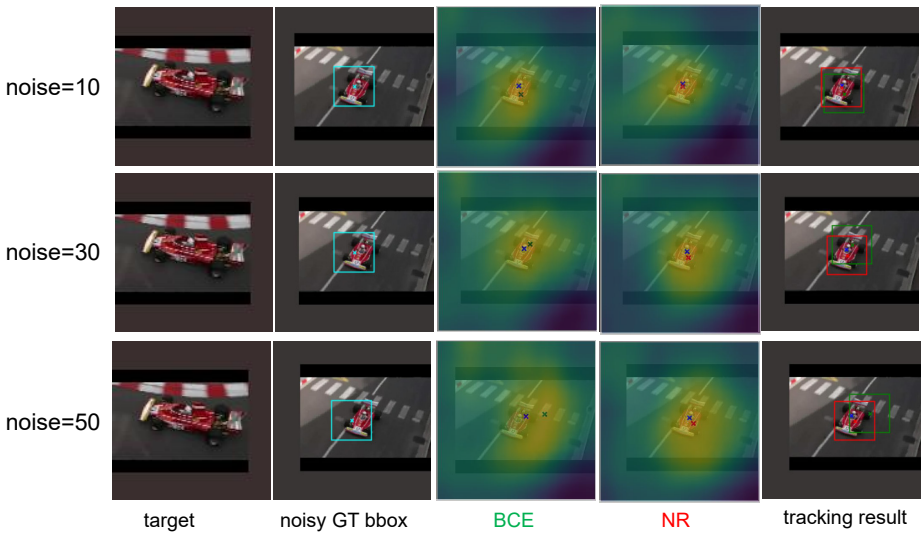

Figure 3: Visualization of response maps and tracking results predicted from models trained with different loss functions and noise levels. The noisy GT bounding box and true GT position are denoted as cyan and blue colors. The results obtained by our proposed NR loss and the original BCE loss are represented as red and green colors.

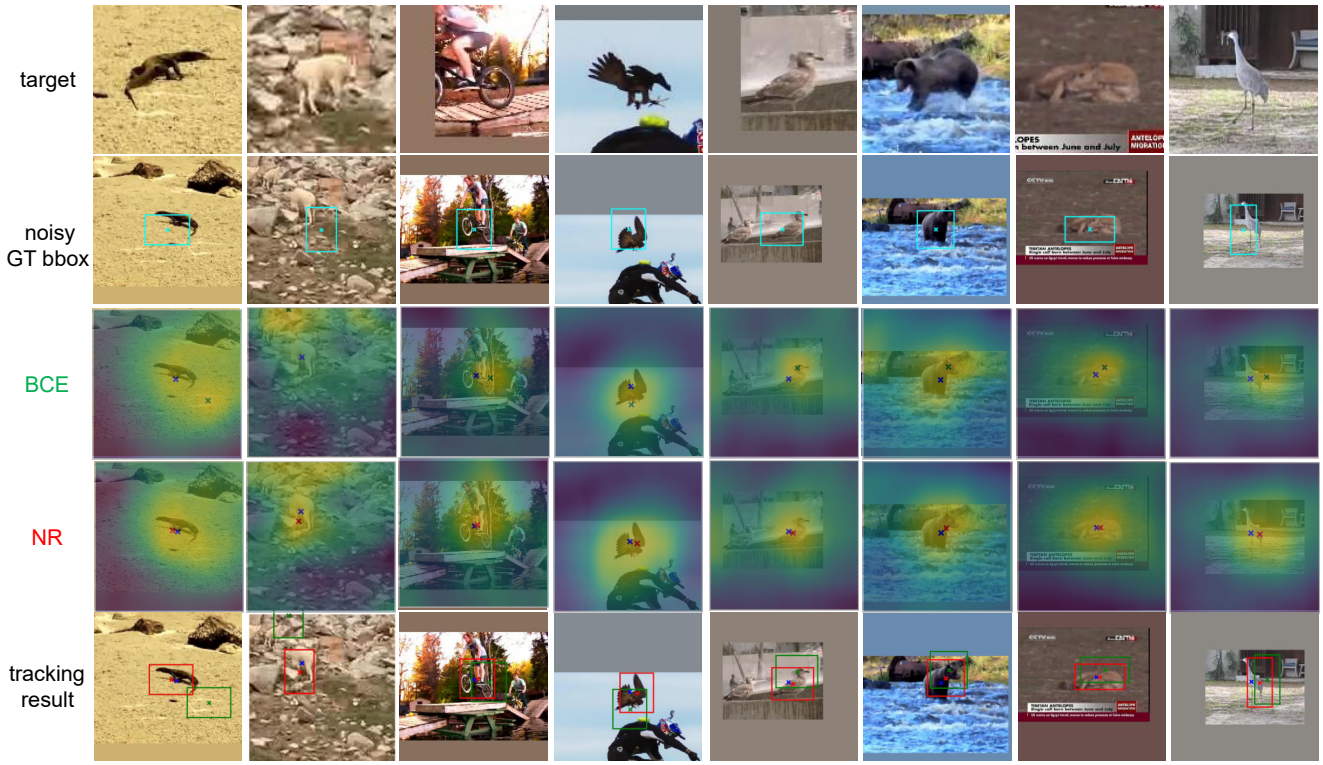

Figure 4: Noise = **50**: visualization of response maps and tracking results predicted from models trained with different loss functions. The noisy GT bounding box and true GT position are denoted as cyan and blue colors. The results obtained by our proposed NR loss and the original BCE loss are represented as red and green colors.

added (Noise=0) to the original training dataset of SiamFC, our noise-robust loss achieves better performance than the BCE loss, mainly because the spatial annotation noise naturally exists in the manually labeled dataset (i.e., ILSVRC-2015 [5]). The performance gap increases as the amount of spatial annotation noise increases. For example, when Noise=50, our NR-loss can outperform the BCE loss with a relative gain of 5.7% in terms of AUC, which demonstrates

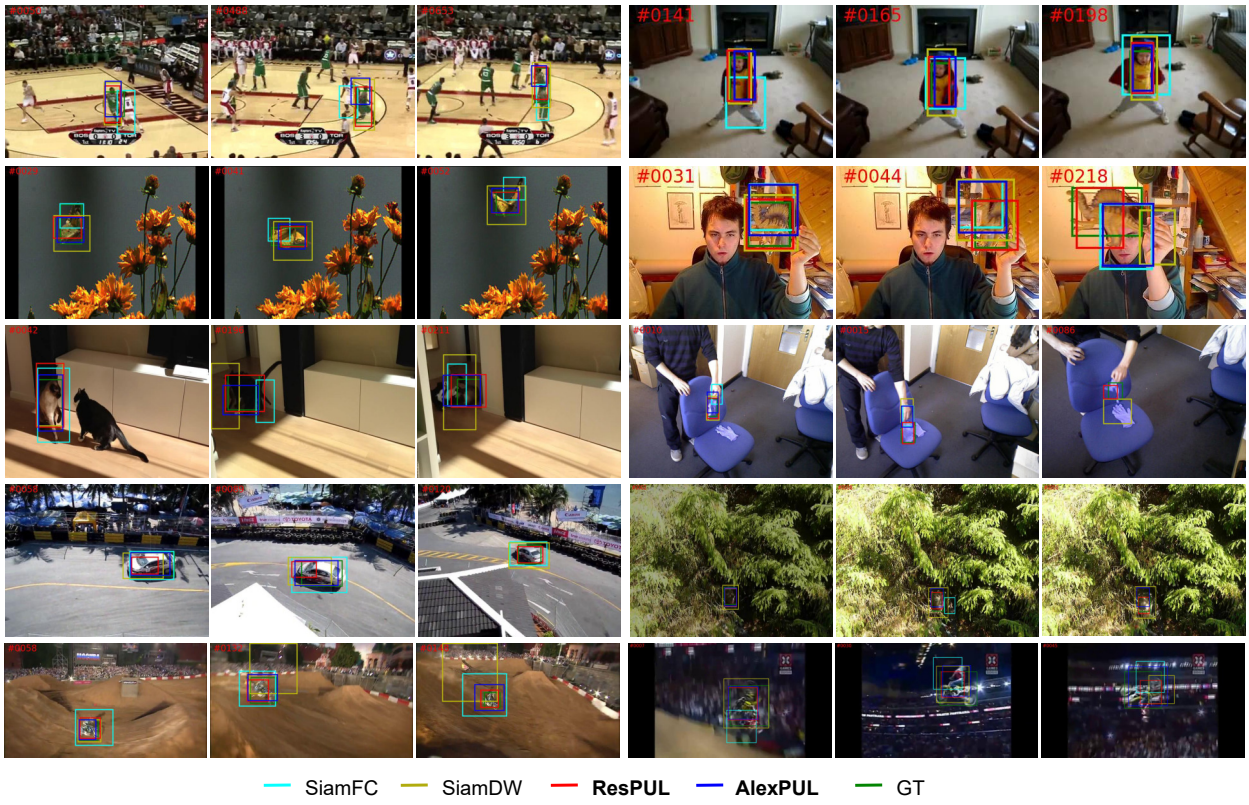

Figure 5: A qualitative comparison of our proposed unsupervised ResPUL and AlexPUL trackers with the supervised baseline trackers on 10 challenging video sequences from the VOT2016 dataset [3]. The sequences from left to right and top to down are *Basketball*, *Blanket*, *Butterfly*, *Dinosaur*, *Fernando*, *Glove*, *Racing*, *Soldier*, *Motocross1* and *Motocross2*.

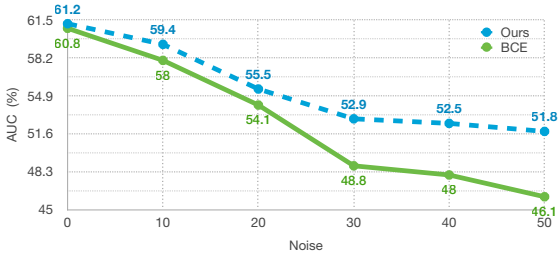

Figure 6: Tracking results (AUC) when training with different amounts of spatial annotation noise (in pixels) using traditional binary cross entropy (BCE) loss and our noise-robust loss.

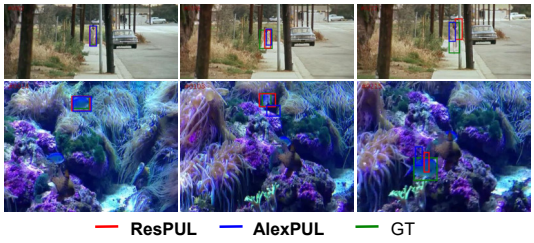

Figure 7: Failure cases of the proposed unsupervised ResPUL and AlexPUL trackers. The two sequences from top to down are *Graduate* and *Fish1* from the VOT2016 dataset.

that our NR-loss better handles the spatial annotation noise.

E. Qualitative Tracking Results

Qualitative comparison. We show the qualitative comparison of our proposed unsupervised ResPUL and AlexPUL trackers with the supervised baseline trackers in Fig. 5. In some cases, our unsupervised trackers can even achieve more accurate online tracking than the supervised baselines. For example, in the *Butterfly* video, SiamDW tends to track a larger region around the butterfly, which also contains the redundant background part, while siamFC only tracks the partial butterfly. In comparison, our AlexPUL and ResPUL trackers can track the butterfly well even the target has large deformation, due to the good generalization ability of the learned unsupervised representations.

Failure cases. Two failure cases of the proposed unsupervised ResPUL and AlexPUL trackers are given in Fig. 7. In the *Graduate* sequence, the man gradually ran closer to the camera with large appearance variations. Our ResPUL tracker fails at the 222-th frame, which is mainly because that the man undergoes significant deformation and mean-

while one similar object appears. This drift problem can be effectively addressed by applying an online updating model. In the *Fish1* sequence, our AlexPUL tracker suffers from the similar failure at the 108-th frame due to the lack of online updating. Additionally, the scale of the fish changes a lot at the 215-th frame, and our trackers cannot accurately estimate the scale. It should be noted that the scale estimation is one main limitation in our baseline trackers SiamFC [1] and SiamDW [6]), since they do not have a scale estimation branch [4] for more accurate bounding box prediction. This problem can be effectively alleviated by applying a scale estimation branch in our baseline trackers.

F. Implementation Details

We discuss additional implementation details in this section. The proposed PUL framework uses several common data augmentation operations to create augmented training samples, including random stretch (i.e., widely used in original SiamFC [1]), random color distortions, random Gaussian blur, random horizon flip and random gray scale. In the temporal correspondence (TC) learning, the contrastive loss is calculated using the template patches (127×127) in the sampled mini-batch patch pairs. The size of the searching patch is the same as the crop size used in original SiamFC [1] and SiamDW [6], which is 255×255.

Our method is implemented using Python 3.7 and PyTorch 1.2.0. The experiments are conducted on a PC with an i7-4.0 GHz CPU and a single GeForce RTX 2080 Ti GPU. The tracking speed of our proposed AlexPUL and ResPUL trackers is the same as our baseline trackers (i.e., SiamFC and SiamDW), which can achieve above real-time speed.

References

[1] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H.S. Vedaldi. Fully-convolutional siamese networks for object tracking. In *ECCVW*, pages 850–865, 2016.

[2] T. Chen, S. Kornblith, and M. Norouzi. A simple framework for contrastive learning of visual representations. In *arXiv:2002.05709*, 2020.

[3] M. Kristan, A. Leonardis, J. Metas, M. Felsberg, R. Pflugfelder, and L. Cehovin. The visual object tracking vot2016 challenge results. In *ICCVW*, pages 777–823, 2016.

[4] B. Li, W. Wu, Z. Zhu, and J. Yan. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018.

[5] O. Russakovsky, J. Deng, and et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[6] Z. Zhang and H. Peng. Deeper and wider siamese networks for raal-time visual tracking. In *CVPR*, 2017.