SiamX: An Efficient Long-term Tracker Using Cross-level Feature Correlation and Adaptive Tracking Scheme

Huajian Huang and Sai-Kit Yeung

***Abstract*— Siamese network based trackers have achieved significant progress in visual object tracking. For the sake of speed, they mainly rely on offline training to learn a mono-level feature correlation between a target template and a search region. During the tracking period, they use a fixed strategy to infer target positions over sequences regardless of target states. However, such approaches are vulnerable in case of long-term challenges e.g. large variance, presence of distractors, fast motion, or target disappearing and the like. In this paper, we propose a new tracking framework, referred to as SiamX, by exploiting cross-level Siamese features to learn robust correlations between the target template and search regions, and also adaptive inference strategies to prevent tracking loss and realize fast target re-localization. Extensive experiments on four benchmarks including VOT-2019, LaSOT, GOT-10k, and TrackingNet show our method significantly enhances the tracker's ability to resist variance and interference, and achieve state-of-the-art results at around 50 FPS.**

I. INTRODUCTION

Visual object tracking is an essential problem which seeks to localize the dynamic target at each frame over a video sequence, given an initial target position. This is still well known as a challenging task in long-term scenarios influenced by various factors like occlusion, deformation, background clutter and etc. A stable and real-time visual tracker has a variety of practical applications, such as augmented reality, autonomous robots, and self-driving cars [1]–[4].

In recent years, Siamese network based methods [5], [6] have attracted great attention because of promising accuracy and speed. The key of Siamese trackers is to capture feature correlations between two branches and then get a response map to identify target positions. The correlations they use are commonly computed from mono-level features extracted by backbone networks. However, as target objects usually experience large deformation, distractors, or other adverse variance, the single feature correlation becomes weaker, which may eventually result in tracking failure.

Additionally, we note that instability of previous methods [6]–[9] partly results from their inference strategy. To increase tracking speed, they perform a local search on images, and the size of the candidate region is directly proportional to the estimated size of the target. As they assume states of the target among consecutive frames are consistent and would not change dramatically, the current search region is always centered at the previously estimated target position, even though the previous estimate is unreliable and the target object has vanished from this local region. Consequently, it is easy to lose tracking when the target is fast-moving, and the system cannot efficiently conduct long-term tracking which requires the ability to deal with out-of-view challenges.

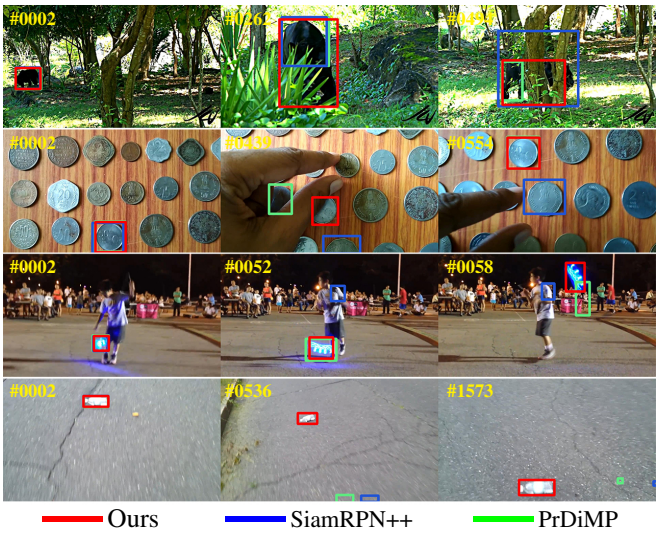


Fig. 1. A qualitative comparison of our method with trackers SiamRPN++ [8] and PrDiMP [10]. Relying on learning robust feature correlations and adaptive tracking scheme, our tracker achieves a better performance in the challenging scenarios, such as larger appearance variance (first row), distractors (second row), fast motion (third row), and temporal disappearance (bottom row).

To address these issues, we propose a novel visual tracking framework learning cross-level feature correlations and an adaptive tracking scheme to enhance performance in long-term challenges, shown in Fig. 2. In deep neural networks, different layers features have distinct characteristics and can provide various meaningful representations of the image [11]. Earlier layers primarily extract low-level features such as color and shape, while deeper layers can provide features with rich semantic information. Therefore, learning correlations between low-level and high-level features can offer comprehensive representations to get robust responses, e.g.,FPN [12]. This benefit to object tracking has not been explored before. In our framework, the correlation fusion module is the bridge to make use of cross-level features. The module leverages distinct features extracted from different layers of backbones to capture robust feature correlations between target and search patches. Furthermore, an anchor-free localization network uses the integrated feature correlation maps to classify targets and directly regress the bounding

Huajian Huang and Sai-Kit Yeung are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. hhuangbg@connect.ust.hk, saikit@ust.hk

This research project is partially supported by an internal grant from HKUST (R9429) and Technology Support Programme of the Innovation and Technology Fund (Ref: ITS/200/20FP).

boxes. As a result, the tracker is able to accurately localize and identify target scales as long as the target is properly in the search region. This inspires and creates conditions for the introduction of the adaptive tracking scheme.

Adaptive tracking scheme is a simple yet effective way to interpret predictions and ensure the tracker searches suitable regions during tracking inference phase. Different from preceding methods, our system drifts search centers in the direction of momentum to reduce the risk of tracking loss caused by fast moving. In addition, the system records target states and continuously computes its probability distribution during the inference phase. When tracking fails or targets inevitably disappear, it will take advantage of statistics and search the most possible region where the target appears. This method enables our tracker to avoid expensive global search and achieves fast re-localization.

Extensive experiments on five benchmarks including VOT-2019 [13], LaSOT [14], GOT-10k [15] and TrackingNet [16] verify the effectiveness of our proposed methods. Our Tracker achieves new state-of-the-art results and also runs at a relatively high speed of 50 FPS.

II. RELATED WORKS

Visual Tracking by Siamese network. The Siamese network encodes two input branches through deep convolution architectures and fuses the encoded features by a specific operation to generate a single output. The output generally represents the relationship between two branches. As visual tracking can be formulated into a problem finding response between object template and the search region, various trackers extend Siamese network for visual tracking. Thus the key problem for Siamese network based trackers is how to learn and make use of robust feature correlations. The pioneering works are SINT [5] and SiamFC [6]. By adding a region proposal network after the Siamese network, SiamRPN [7] significantly enhances the performance of classical Siamese tracking, but it requires carefully pre-designed anchor boxes. The follow-up work [17] develops distractor-aware offline training to enhance capability to withstand the impact of distractors. Ocean [9] is an object-aware anchor-free tracking framework and relies on an online branch to attain better results. Learning online and updating target features indeed increase trackers' accuracy, but the computational cost also has a noticeable increase. To take advantage of deep neural network power, SiamRPN++ [8] and SiamBAN [18] employ multiple predicted heads on distinct layers of the backbone and improve accuracy via aggregating overall predictions. But as the network becomes deeper and multiple regression heads are used, the tracking speed has been severely impaired, e.g.,SiamRPN (160FPS)→SiamRPN++ (35FPS). In contrast, although we only use a regression head, we strive to generate comprehensive representations between targets and search regions by learning robust cross-level feature correlations, which enables our tracker to outperform state-of-the-art trackers, even including discriminative correlation filter based methods (e.g.,PrDiMP [10]).

Visual Tracking Inference Scheme. Tracking inference scheme is the strategy to interpret network prediction and perform tracking over sequences in the inference phase. Traditional tracking frameworks rely on multi-scale search to identify target scales and aspect ratios. Since the introduction of object detector modules, current Siamese network based trackers [7]–[9] generally formulate visual tracking task as a local one-shot detection task. They assume target states would not have large variance over sequences so that trackers search for targets within square regions centered at previous positions. They also use a fixed cosine window to suppress distractors and large displacement before proposal selection. However, once the basic assumption was violated, their performance would significantly degrade. The degradation becomes even more severe in long-term tracking scenarios where out-of-view situations such as occlusion and tracking loss become inevitable. The estimates become highly unreliable given that the search regions are wrong and the system cannot re-track the target unless it proactively reappears within the regions. DaSiamRPN [17] applies a local-to-global searching strategy for target re-localization, but it is unstable in case of large scale variance and presence of distractors.

III. SIAMX NETWORK ARCHITECTURE

As illustrated in Fig. 2, the proposed network architecture is composed of feature extraction backbones, a cross-level feature correlation fusion module and an anchor-free localization network. The feature extraction network utilizes share-parameter backbones to extract multiple features of target templates and search patches. And then, the feature correlation fusion module contributes to learning cross-level correlations and integrates them by the optimized weights. The robust correlation enhances the framework's ability to resist variance and interference. Finally, based on correlation maps, the anchor-free localization network classifies target and regresses bounding boxes, achieving stable tracking.

A. Siamese Feature Extraction

Here, we adopt a modified Resnet-50 [19] as the feature extraction backbone. Compared to the vanilla Resnet-50, only the first four convolution blocks ($conv1,...,conv4$) are retained. In order to retain detailed spatial information and widen the receptive field, all of the 3×3 convolution kernels in the fourth block are dilated [20] at a rate of 2, and the convolution stride of the down-sampling operation is reduced from 2 to 1. The input of the Siamese feature extraction network is an image pair, i.e.,template and search patch, representing the object of interest and the candidate region respectively. To reduce computational costs, they are resized to specific sizes and the search patch is double the size of the template patch. Both patches are encoded into deep feature maps by fully convolutional neural networks with the same parameters. Finally, the number of feature channels is decreased to 256 through a bottleneck layer. For convenience, we let $\mathbf{f}_z^{conv\,i}$ and $\mathbf{f}_x^{conv\,i}$ denote the template (z) and search patch (x) features extracted from specific *conv* blocks respectively, where $i=1,...,4$.

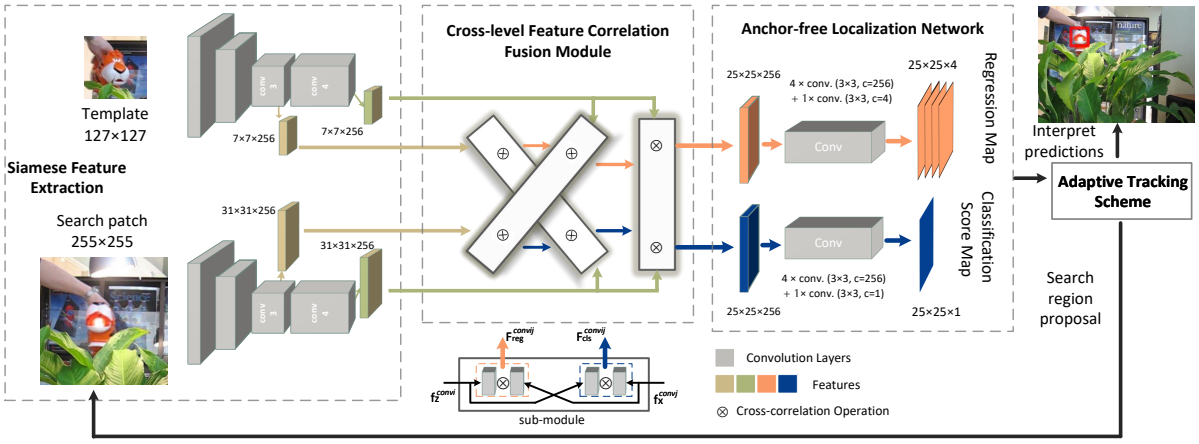


Fig. 2. The proposed tracking framework SiamX consists of four major components: 1) a Siamese feature extraction network, 2) a cross-level feature correlation fusion module involving three identical sub-modules to transfer extracted features into classification and regression branches and learn the optimal cross-level correlations between the target template and search patch, 3) an anchor-free network to localize target and regress boundaries, and 4) an adaptive tracking scheme to track dynamic objects spatially and temporally.

B. Cross-level Feature Correlation

The cross-level feature correlation fusion module, illustrated in Fig. 2 (middle), takes responsibility to learn robust feature correlations between target and search patches. In a deep convolution neural network, different layer features have distinct characteristics [12]. The features extracted from earlier layers can provide fine-grained information, like object shapes. As the layers increase, the extracted features capture abstract semantic information that is scale-invariant. In other words, shallow features can contribute to object localization, while deep-level features can enhance discriminability. The correlations between low-level features and high-level features can offer additional representations to improve framework robustness, as Fig. 3 shows.

We define mono-level feature correlation as the correlation between two features at the same level, and bi-level feature correlation as the correlation between two different-level features. The proposed module consists of three sub-modules with the same network architectures to calculate two bi-level feature correlation maps and one mono-level feature correlation map. The inputs of each sub-module are particular features containing the target and search patch. Each sub-module firstly projects features into classification and regression branches by parallel convolution layers, and then utilizes depth-wise cross-correlation operations [8] to generate feature maps that represent the correlations between inputs. Specifically, we will calculate the correlation between $\mathbf{f}_z^{conv3}$ and $\mathbf{f}_x^{conv4}$, the correlation between $\mathbf{f}_z^{conv4}$ and $\mathbf{f}_x^{conv3}$, and the correlation between two *conv*4 features, $\mathbf{f}_z^{conv4}$ and $\mathbf{f}_x^{conv4}$. This process in each submodule can be depicted as:

$$\mathbf{F}_{reg}^{convij} = \Phi_{reg}^{z}(\mathbf{f}_z^{convi}) \otimes \Phi_{reg}^{x}(\mathbf{f}_x^{convj}), \tag{1}$$

$$\mathbf{F}_{cls}^{convij} = \Phi_{cls}^{z}(\mathbf{f}_z^{convi}) \otimes \Phi_{cls}^{x}(\mathbf{f}_x^{convi}), \tag{2}$$

where $\mathbf{F}_{reg}^{convij}$ and $\mathbf{F}_{cls}^{convij}$ represent the feature correlations in regression and classification branches respectively, $(i,j) \in [(3,4),(4,3),(4,4)]$, Φ represents a 3×3 convolution layer, and $\otimes$ denotes cross-correlation operation. Finally, according to the trained weights, these correlation maps in the two branches respectively are aggregated to form two independent correlation feature maps:

$$\mathbf{F}_{reg} = \sum_{convij} w_{reg}^{convij} \mathbf{F}_{reg}^{convij}, \tag{3}$$

$$\mathbf{F}_{cls} = \sum_{convij} w_{cls}^{convij} \mathbf{F}_{reg}^{convij}. \tag{4}$$

The correlation maps will become the inputs of the anchor-free localization network and contribute to precise classification and regression.

C. Anchor-free Localization

The anchor-based or anchor-free model is the concept in the field of object detection. As the name implies, the anchor-free model does not rely on pre-defined candidate boxes to detect and classify the objects, which is more flexible. FCOS [21] is a representative anchor-free detector and contains three distinct heads, including one regression head and two centerness/classification heads. Since visual tracking is only required to classify target and non-target, we only adopt the regression head and the centerness branch, as illustrated in Fig. 2 (right). Both heads consist of 5 convolution layers with 3×3 kernels. The centerness branch produces a single-channel map of classification scores, denoted by $|\mathbf{P}_{cls}|_{w\times h\times1}$. The positional correspondence between the search patch S and the prediction map P is

$$\begin{aligned} S_x &= \frac{S_w}{2} + (P_x - \frac{w}{2}) \times s, \\ S_y &= \frac{S_h}{2} + (P_y - \frac{h}{2}) \times s, \end{aligned} \tag{5}$$

where (S_x, S_y) and (P_x, P_y) represent the coordinates of sampling pixels in the search patch and the prediction map, respectively, (S_w, S_h) and (w,h) denote the width and height of the search patch and the prediction map, respectively, while s denotes the down-sampling stride.

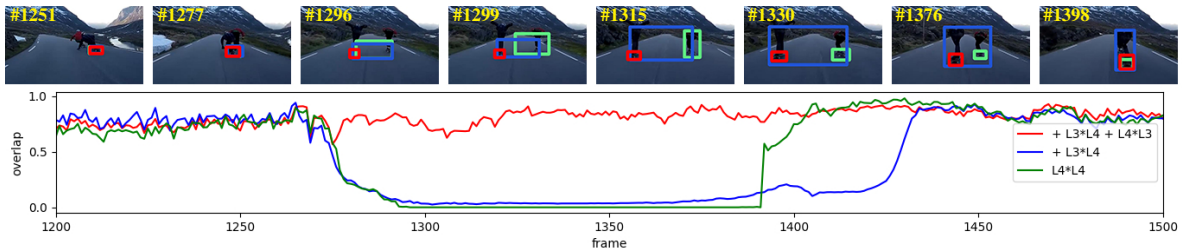


Fig. 3. The cross-level correlation can increase framework's ability against interference and variance. Li*Lj denotes the correlation between $\mathbf{f}_z^{conv\,i}$ and $\mathbf{f}_x^{conv\,j}$. Horizontal axis: the frame number. Vertical axis: the overlap between ground truth and tracking results.

The regression head directly derives the distances from the pixels to the four sides of the object boundaries, generating a repression map of 4 channels, denoted by $\left|\mathbf{P}_{reg}\right|_{w\times h\times 4}$. For convenience, we use $O=(l,t,r,b)$ to denote the offsets from a pixel to the left, top, right and bottom boundaries of the target. We also use the coordinates of top-left and bottom-right corners to represent the rectangular bounding box $B=(S_{x_0},S_{y_0},S_{x_1},S_{y_1})$. The bounding box B can be computed by

$$\begin{aligned} S_{x_0}&=S_x-l, S_{y_0}=S_y-t,\\ S_{x_1}&=S_x+r, S_{y_1}=S_y+b. \end{aligned} \tag{6}$$

D. Loss Function

The original annotation is the bounding box of the target object in an image and denoted by $B^*=(S^*_{x_0},S^*_{y_0},S^*_{x_1},S^*_{y_1})$. To get the regression ground-truth, we first use Eq. (6) to compute the offsets of each pixel, $O^*=(l^*,t^*,r^*,b^*)$, where $O^*\in\left|\mathbf{O}^*\right|_{S_w\times S_h\times 4}$. Utilizing Eq. (5) to down-sample the $\mathbf{O}^*$, we can obtain final training ground-truths for the regression, $\left|\mathbf{P}^*_{reg}\right|_{w\times h\times 4}$. Moreover, the pixels falling in the B^* are assigned with positive labels, while the outside pixels are assigned with negative labels and will be ignored during training. Since the task of the classification branch is to predict target centers, we label the pixels close to the center as 1. Thus, the training labels for classification are

$$\mathbf{P}^*_{cls}(P_x,P_y)=\begin{cases}1, & \text{if } d((P_x,P_y),c^*)\le r\\ 0, & Otherwise\end{cases}, \tag{7}$$

where r is a hyper-parameter, and the ground-truth center $c^*=(P^*_{x_c},P^*_{y_c})$ can be projected from the target annotation center in the search image, $S^*_c=(S^*_{x_c},S^*_{y_c})=\left[(S^*_{x_0}+S^*_{x_1})/2,(S^*_{y_0}+S^*_{y_1})/2\right]$, by Eq. (5). We utilize intersection over union (IoU) loss [22] and binary cross-entropy (BCE) loss [23] to jointly optimize regression and classification branch. Therefore, the loss function is defined as

$$L=\lambda_0 L_{reg}+\lambda_1 L_{cls}\,, \tag{8}$$

where λ_0 and λ_1 are two trade-off hyper-parameters.

E. Offline Training

The proposed network is trained with image pairs, using one image as the target template and the other as the search patch to evaluate the framework. Training pairs are randomly sampled from the training splits of COCO [24], GOT-10k [15], LaSOT [14], and TrackingNet [16]. To simulate intricate tracking scenarios, we apply random jitters to the sample centers and the sample sizes. Before being fed to the network, training pairs would be resized to 127×127 and 255×255 respectively. We use synchronized stochastic gradient descent (SGD) and train on 6 GPUs for 50 epochs. Each GPU processes 32 pairs of images per iteration. The backbone network, ResNet-50 [19], is initialized with the parameters pre-trained on ImageNet [25] and frozen in the first 10 epochs. The warmup learning rate used in the first 5 epochs is from 0.001 to 0.005, while the learning rate exponentially decays from 0.005 to 10^{-5} in the remaining epochs. Weight decay and momentum are set at 0.0001 and 0.9 respectively. Since we only take advantage of the first four blocks of ResNet-50, the stride s in Eq. (5) is 8, the radius r in Eq. (7) is 2, the weight parameters λ_0 and λ_1 in Eq. (8) are simply set to 1.

IV. ADAPTIVE TRACKING SCHEME

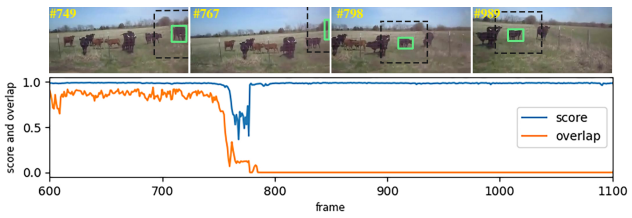
To further enhance tracker performance in long-term tracking scenarios, we exploit an adaptive tracking scheme. During tracking phase, target positions (S^t_c), displacements d^t and classification scores ($conf^t$) will be recorded. As we use *sigmoid* as the activation function in the classification branch, the range of classification scores is between 0 and 1. We can adapt the scores as confidence coefficients of estimates to reflect tracking quality.

Momentum Compensation. We perform local search to avoid computational costs increases which is similar to previous methods, but we no longer assume target states among nearby frames always have small changes. If the target displacements between consecutive frames exceeds target sizes, it considers the target object is at a fast-moving state. To avoid targets leaving the search regions, the search center drifts in the direction of momentum:

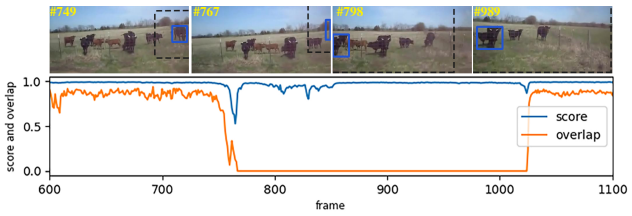
$$\begin{aligned} d^{t-1}&=S^{t-1}_c-S^{t-2}_c\,,\\ (S^t_c)'&=S^{t-1}_c+d^{t-1}\cdot\xi\,(conf^{t-1}-th)\,, \end{aligned} \tag{9}$$

Where S^{t-1}_c and S^{t-2}_c are previous target positions, $(S^t_c)'$ is the predicted position, ξ denotes a unit step function, while $conf^{t-1}$ denotes confidence coefficients of the last estimate. th is set as 0.85. Momentum compensation can reduce the risk of tracking failure caused by fast motion.

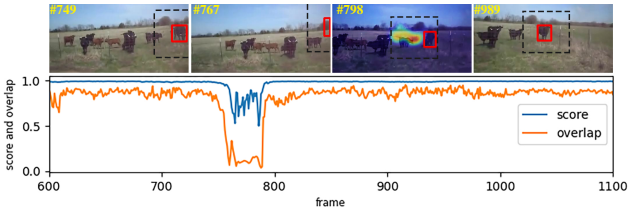
Target Re-localization. An efficient re-localization algorithm is important in long-term tracking. When the classification score is lower than the threshold, the estimate becomes



(a) Fixed search scheme



(b) Local-to-global search



(c) Adaptive tracking scheme

Fig. 4. Tracking results of using different strategies in out-of-view scenarios. The blue line represents the classification score while the orange line represents the IoU between ground truth and tracking results. On each snapshot, the rectangle denotes target estimates while dashed lines denote search boundaries. Adaptive tracking enables the tracker to return normal quickly.

unreliable and the target is considered to have vanished from the search area. If we persist in this region, it cannot re-localize the target unless it proactively appears. However, the size of the local search region is only proportional to the target size. When the target is tiny relative to the image, the probability of target occurrence within the local region is low. This means the system cannot return to normal for a long time. On the other hand, if we perform a global search [17] for a small target, it has difficulty in aspect ratio estimates as the framework cannot learn such extreme scenes during offline training. These methods would also be fragile in case of background clutter and distractors.

Different from previous approaches, we seek a better estimate of the target by searching the most likely region where the target reappears. Supposing the estimates of the target position are Gaussian, we compute the probability distribution of the target and maintain a heat-map:

$$H(T) = \sum_{t=1}^{T} p(t)\xi(conf^t - th), \tag{10}$$

where $p(t) \sim \mathcal{N}(\boldsymbol{S}_c^t, \boldsymbol{\Sigma}^t)$, obeys bi-variate normal distribution, S_c^t is the target position, and the kernel size is associated with the target size. When the target disappears and tracking loses, according to statistics, $argmax\{H(T)\}$, it would search the most possible region for a reliable estimate which allows our tracker to achieve fast re-localization.

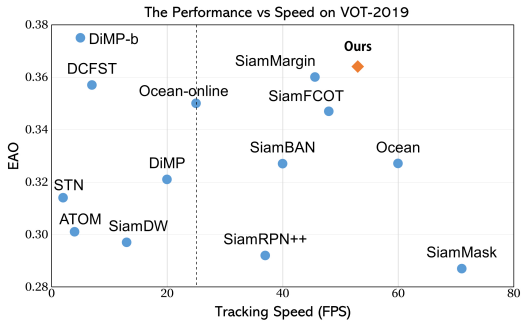
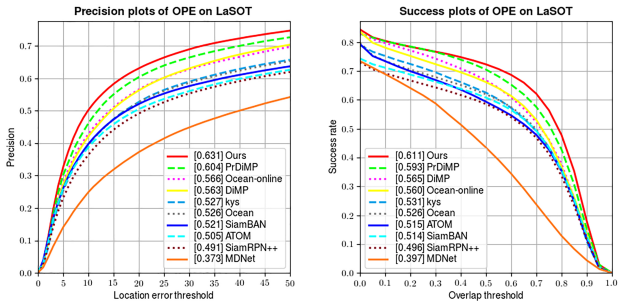


Fig. 5. A comparison of expected average overlap (EAO) and tracking speed on VOT-2019.



(a) Precision Plot (b) Success Plot

Fig. 6. Success and precision plots on LaSOT.

V. EVALUATION

Our approach is implemented in Python using PyTorch and runs at more than 50 FPS on a PC with Intel i7-6700k CPU and a single Nvidia GTX 1080 GPU. We extensively compare the proposed tracker SiamX with state of the art. The results are presented in Fig. 1, 5, 6 and Table I - II.

A. State-of-the-art Comparison

VOT-2019 [13] contains 60 sequences and the official toolkit evaluates performance in terms of accuracy, robustness, and expected average overlap (EAO). As Fig. 5 shows, our tracker achieves 0.364 EAO score and great balance between accuracy and speed.

LaSOT [14] is a high-quality long-term benchmark with 1400 video sequences in total, 280 of which form the test set for evaluation. Each sequence has on average 2,500 frames. The results on LaSOT are illustrated in Fig. 6. Our tracker has the top performance and gains an AUC score of 0.611, significantly outperforming PrDiMP [10] and DiMP [29] by 1.8% and 3.6% respectively.

GOT-10k [15] is a high-diversity database for generic object tracking. The test set has 180 video segments covering various object and motion classes and its annotation is sequestered. They provide an online server to fairly compare trackers and evaluate success rates (SR) and average overlap (AO) scores. The results are shown in Table I. It can be seen that our proposed tracker achieves the best AO score, 0.638. Although the gaps in terms of AO score among top-3 trackers are modest, the inference speeds of PrDiMP [10] and KYS [30] are 2 to 3 times slower than ours. Compared

TABLE I

STATE-OF-THE-ART COMPARISON ON THE GOT-10K TEST SET. THE METRICS INCLUDE AVERAGE OVERLAP (AO), AND SUCCESS RATES (SR) AT THE OVERLAP THRESHOLDS 0.50 AND 0.75. **Red**, BLUE AND GREEN HIGHLIGHT THE TOP-3 TRACKERS.

Tracker	GOTURN [26]	SiamFC [6]	CFNet [27]	SiamRPN++ [8]	ATOM [28]	DiMP [29]	Ocean [9]	Ocean^o [9]	PrDiMP [10]	KYS [30]	**SiamX** (Ours)
$SR_{0.50}$	0.375	0.353	0.404	0.618	0.634	0.717	0.738	0.751	0.695	0.721	**0.757**
$SR_{0.75}$	0.124	0.098	0.144	-	0.402	0.492	0.465	0.473	**0.543**	0.515	0.531
AO	0.347	0.348	0.374	0.518	0.556	0.611	0.592	0.611	0.634	0.636	**0.638**
FPS	-	86	41	35	30	43	55	25	30	20	50

TABLE II

A COMPARISON ON TRACKINGNET IN TERMS OF PRECISION, NORMALIZED PRECISION, AND AREA UNDER CURVE (AUC) SCORE.

Tracker	MDNet [31]	UPDT [32]	DaSiamRPN [17]	SiamRPN++ [8]	ATOM [28]	SPM [33]	SiamAttn [34]	DiMP [29]	PrDiMP [10]	KYS [30]	**SiamX** (Ours)
Precision	0.565	0.557	0.591	0.694	0.648	0.661	-	0.687	0.704	0.688	**0.723**
Norm. Prec.	0.705	0.702	0.733	0.800	0.771	0.778	**0.817**	0.801	0.816	0.800	0.813
AUC	0.606	0.611	0.638	0.733	0.703	0.712	0.752	0.740	0.758	0.740	**0.760**

TABLE III

QUANTITATIVE COMPARISON RESULTS OF THE PROPOSED FRAMEWORK AND ITS VARIANT WITH DIFFERENT FEATURE CORRELATION MODULES ON LASOT BENCHMARK. Li⊗Lj REPRESENTS THE CORRELATION BETWEEN $\mathbf{f}_z^{conv\,i}$ AND $\mathbf{f}_x^{conv\,j}$.

L4⊗L4	L4⊗L3	L3⊗L4	L3⊗L3	AUC
✓				0.564
	✓			0.555
		✓		0.545
			✓	0.536
	✓	✓		0.569
✓	✓			0.587
✓		✓		0.583
✓			✓	0.581
✓	✓	✓	✓	0.593
✓	✓	✓		0.594

to tracker DiMP [29] and Ocean-online [9], SiamX achieves a relative gain of 2.7% in term of AO score.

TrackingNet [16] is a large-scale dataset and benchmark. The 511 sequences in the test set cover a wide category of tracking objects in broad and diverse backgrounds. As reported in Table II, our approach attains the top tracking precision among the compared methods. The precision score of SiamX surpasses that of PrDiMP [10] and KYS [30] by 1.9% and 3.5% respectively. Moreover, SiamX also achieve a top AUC score, 0.760, and considerably outperforms SiamRPN++ [8] and DiMP [29] which have AUC scores of 0.733 and 0.740, respectively.

B. Ablation Study

To verify the effect of the proposed framework and adaptive tracking scheme, we conduct the ablation study on the large-scale benchmarks LaSOT [14].

Cross-level Feature Correlation Fusion Module. To analyze the contributions of different feature correlations, we disabled the adaptive tracking scheme and performed a component-wise analysis on this module, and the results are shown on the Table III. When only a single correlation map is used, the mono-level feature correlation map, L4⊗L4 , gains the highest AUC score. The combination of multiple correlation maps indeed improves success ratios. Our proposed module achieves top performance, which computes and aggregates three distinct correlations, including one mono-level feature correlation map, L4⊗L4, and two bi-level feature correlation maps, L4⊗L3 and L3⊗L4. Adding another correlation map L3⊗L3 leads to overfitting.

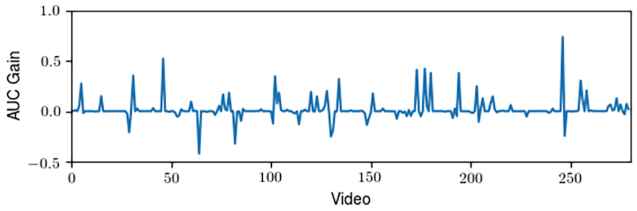


Fig. 7. AUC gains at each video of LaSOT benchmark after deploying the adaptive tracking scheme on our tracker.

Adaptive Tracking Scheme. The adaptive tracking scheme reduces the risk of tracking loss causing by fast motion and enables the tracker to refine estimates and re-localize the target. In our tracker, the adaptive tracking scheme brings a clear gain and the AUC score rises by 1.7% and reaches 0.611 finally on LaSOT benchmark. Conversely, the global search strategy would decrease the score by 0.01. In addition, we introduced the adaptive tracking scheme to another anchor-free tracker, Ocean [9]. The success ratio of Ocean considerably grows from 0.526 to 0.546 and the performance gains in both frameworks are similar. This actually reflects the generalization and efficacy of our proposed method. However, we also found it sometimes would impair accuracy. As Fig. 7 shows, the AUC scores of 4% videos suffer a slightly decrease which may result from improper distribution estimates. We will explore the specific cases and further improve the tracking scheme as future work.

VI. CONCLUSION

In this paper, we propose a novel visual tracking framework named SiamX taking advantage of cross-level feature correlations between two branches to increase discernment and an adaptive tracking scheme to enhance tracker capabilities in dealing with long-term scenarios. The extensive experiments clearly demonstrate the effectiveness and efficiency of our proposed methods. Our tracker achieves state-of-the-art results and runs at around 50 FPS.

REFERENCES

[1] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep learning for visual tracking: A comprehensive survey," *arXiv preprint arXiv:1912.00535*, 2019.

[2] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *PAMI*, vol. 36, no. 7, pp. 1442–1468, 2013.

[3] H. Uchiyama, T. Taketomi, S. Ikeda, and S. Mori, "Basics and advances in monocular vslam," in *Smart Sensors and Systems*. Springer, 2020, pp. 93–104.

[4] H. Huang, W.-Y. Lin, S. Liu, D. Zhang, and S.-K. Yeung, "Dual-slam: A framework for robust single camera navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4942–4949.

[5] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *CVPR*, 2016, pp. 1420–1429.

[6] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV*, 2016, pp. 850–865.

[7] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *CVPR*, June 2018.

[8] 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*, June 2019.

[9] J. F. B. L. W. H. Zhipeng Zhang, Houwen Peng, "Ocean: Object-aware anchor-free tracking," in *ECCV*, August 2020.

[10] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *CVPR*, 2020, pp. 7183–7192.

[11] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *ICCV*, 2015, pp. 3074–3082.

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[13] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Cehovin Zajc, O. Drbohlav, A. Lukezic, A. Berg, *et al.*, "The seventh visual object tracking vot2019 challenge results," in *ICCV Workshops*, 2019.

[14] 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, pp. 5374–5383.

[15] 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.

[16] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *ECCV*, 2018, pp. 300–317.

[17] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *ECCV*, September 2018.

[18] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *CVPR*, 2020, pp. 6668–6677.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[20] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[21] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *ICCV*, 2019, pp. 9627–9636.

[22] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 516–520.

[23] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

[24] 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*. Springer, 2014, pp. 740–755.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[26] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *ECCV*. Springer, 2016, pp. 749–765.

[27] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *CVPR*, 2017, pp. 2805–2813.

[28] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *CVPR*, 2019, pp. 4660–4669.

[29] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *ICCV*, 2019, pp. 6182–6191.

[30] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Know your surroundings: Exploiting scene information for object tracking," in *ECCV*, 2020.

[31] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016, pp. 4293–4302.

[32] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *ECCV*, 2018, pp. 483–498.

[33] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "Spm-tracker: Series-parallel matching for real-time visual object tracking," in *CVPR*, 2019, pp. 3643–3652.

[34] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable siamese attention networks for visual object tracking," in *CVPR*, 2020, pp. 6728–6737.