END-TO-END TEMPORAL FEATURE AGGREGATION FOR SIAMESE TRACKERS

Zhenbang Li[a,c], *Qiang Wang*[a,c], *Jin Gao*[a], *Bing Li*[a,*], *Weiming Hu*[a,b,c]

[a]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b]CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing, China
[c]University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

While siamese networks have demonstrated the significant improvement on object tracking performances, how to utilize the temporal information in siamese trackers has not been widely studied yet. In this paper, we introduce a novel siamese tracking architecture equipped with a temporal aggregation module, which improves the per-frame features by aggregating temporal information from adjacent frames. This temporal fusion strategy enables the siamese trackers to handle poor object appearance like motion blur, occlusion, *etc*. Furthermore, we incorporate the adversarial dropout module in the siamese network for computing discriminative target features in an end-to-end-fashion. Comprehensive experiments demonstrate that the proposed tracker performs favorably against state-of-the-art trackers.

***Index Terms*—** Visual object tracking, siamese network, feature aggregation, adversarial training

1. INTRODUCTION

Visual object tracking is the task of estimating the state of an arbitrary target in each frame of a video sequence. Recently, siamese networks have demonstrated the significant improvement on object tracking performances. However, the learned generic representation may be less discriminative because of the deteriorated object appearances in videos (Fig. 1), such as motion blur, occlusion, *etc*. Researchers try different ways to improve the feature representation. For example, SA-Siam [1] separately trains two branches to keep the heterogeneity of semantic/appearance features. In DaSiamRPN [2], a novel distractor-aware incremental learning module is designed, which can effectively transfer the general embedding to the current video domain and incrementally catch the target appearance variations during inference. SiamRPN++ [3] introduces a simple yet effective sampling strategy to drive the siamese tracker with more powerful deep architectures. These efforts have produced some impact and improved state-of-the-art accuracy. However, all above siamese algorithms perform tracking based on features cropped from only the current frame, which limits the power of siamese trackers.

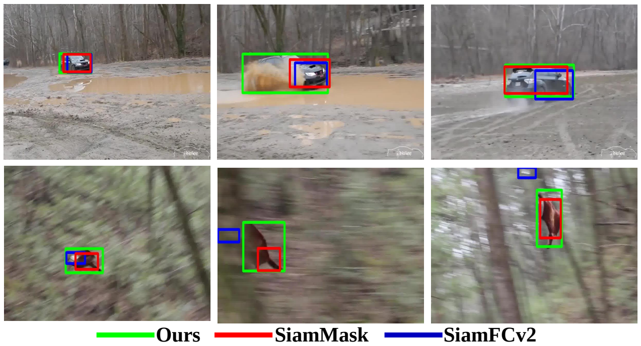

Fig. 1. A comparison of our method with the stage-of-the-art trackers SiamMask and SiamFCv2 in challenging situations. The example frames are from the GOT-10k testing set. Our approach effectively handles poor object appearance compared to existing approaches.

Actually, the video has rich information about the target and such temporal information is an important basis for video understanding and tracking. For example, in video object detection, FGFA [4] leverages temporal coherence on feature level. It improves the per-frame features by aggregation of nearby features along the motion paths, and thus improves the video recognition accuracy. In video object segmentation, STCNN [5] introduces a temporal coherence module, which focuses on capturing the dynamic appearance and motion cues to provide the guidance of object segmentation. In discriminative correlation filter-based object tracking, FlowTrack [6] focuses on making use of the rich flow information in consecutive frames to improve the feature representation and the tracking accuracy. However, how to utilize the temporal information in siamese trackers has not been widely studied yet.

In this paper, we aim to take full advantage of temporal information in siamese trackers. We introduce a novel siamese tracking architecture equipped with a temporal aggregation module, which improves the per-frame features by aggregating features from adjacent frames. This temporal fusion strategy enables the siamese tracker to handle poor object appearance like motion blur, occlusion, *etc*. To achieve this, we shift the channels along the temporal dimension [7] in the backbone of the siamese network. Note that features of the same

*Corresponding author.

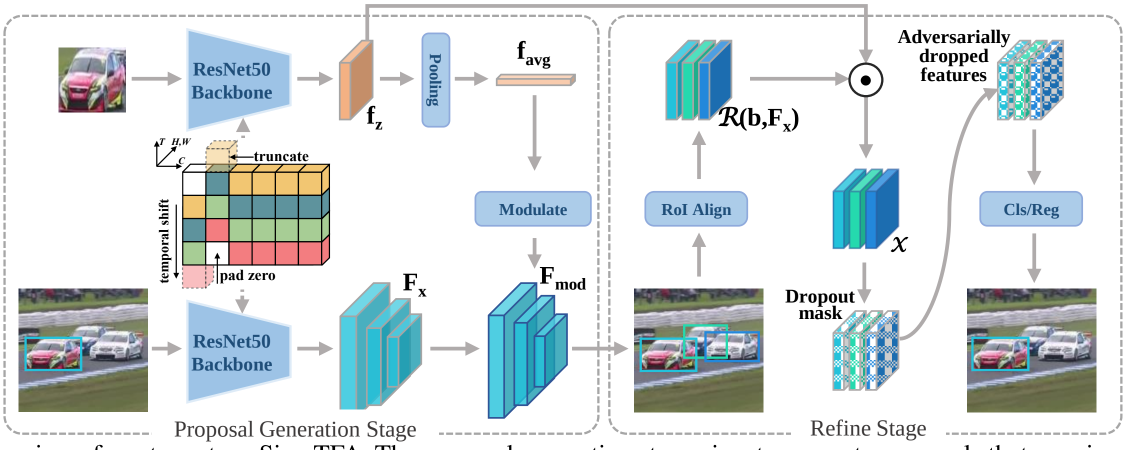

Fig. 2. Overview of our two-stage SiamTFA. The proposal generation stage aims to generate proposals that are visually similar to the given template target. In this stage, we introduce a temporal aggregation module to utilize the temporal information. The refine stage aims to select the target from candidate proposals. In this stage, we insert an adversarial dropout module to learn more robust feature.

object are usually not spatially aligned across frames due to video motion [4], so the temporal shift is only performed on the residual layers [7] to preserve the spatial feature learning capability of the siamese tracker. Different from other temporal fusion methods [8, 9], the proposed method is able to be trained end-to-end on larger-scale datasets. Additionally, our temporal fusion method is easy to implement, without changing the siamese tracking architecture or using optical flow [6].

To improve the robustness of target features, we further incorporate an adversarial dropout [10] module in the siamese tracking network. Specifically, we first predict adversarial dropout masks based on divergence maximum. Then, we aim to minimize the divergence between the randomly dropped features and the adversarially dropped features. This module has both the advantages of dropout and adversarial training: the dropout makes our siamese network randomly disconnects neural units during training to prevent the co-adaptation of target features and the adversarial training enforces our tracker to learn difficult cases.

2. THE PROPOSED METHOD

In this section, we will introduce the proposed siamese architecture-based tracking method, namely SiamTFA (Fig. 2), which is inspired by the great success of siamese trackers [3, 11]. Specifically, SiamTFA takes an image pair as input, comprising a template image and a search image. The template image is the image patch cropped from the initial frame according to ground truth bounding box. The search image is one whole frame in the remaining of the video. Both inputs share the same feature extractor and parameters. Inspired by the success of the two stage detection paradigm [12], our siamese tracker is also a two stage method. The first stage aims to generate proposals that are visually similar to the given template target. In this stage, we introduce a temporal aggregation module to enhance the temporal information (Sec. 2.1). The second stage aims to identify the target from candidate proposals. In this stage, we insert an adversarial dropout module to learn more robust features (Sec. 2.2).

2.1. Temporal aggregation module

The proposal generation stage consists of 3 components: (1) feature extractor, (2) temporal aggregation module, and (3) feature modulation module. The feature extractor generates the search features and the template feature for the search image and the template image, respectively. The temporal aggregation module is integrated into the feature extractor to utilize the temporal information. The feature modulation module merge the search features and the template feature to recognize the candidate targets.

Feature extractor To deal with the scale change of the target, we use Res50-FPN [13] as our feature extractor. **F**eature **P**yramid **N**etwork (FPN) exploits the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. Our siamese FPN takes a template image and a search image as input. For the search image, the FPN outputs proportionally sized feature maps at multiple levels, in a fully convolutional fashion. We denote the output for the search image as $F_x = \{f_x^i\}_{i=1:5}$, and note that they have strides of {4, 8, 16, 32, 64} pixels with respect to the input search image. For the template image, we use the last stage of the FPN output as the template feature with a spatial size of 7 × 7.

Temporal aggregation module Most popular siamese trackers [3, 11] use the still image to make prediction. This limits the ability of these siamese trackers. On one hand, tracking on single frame generates unstable results and fails when appearance is poor (Fig. 1); on the other hand, temporal adjacent frames can provide more information about the target. So we aim to improve the per-frame features by aggregating features of adjacent frames. Specifically, we insert a temporal aggregation module into the last stage of the feature extractor. To model temporal information, the images in one batch are several adjacent frames in the same video and are sorted by time, so we can regard the batch dimension as the time dimension. Assume the feature map at the last stage of the feature extractor is $f \in \mathbb{R}^{T\times C\times H\times W}$. For each time $t \leq T$, we first split feature $f^t \in \mathbb{R}^{C\times H\times W}$ into 3 parts along the channel dimension: $f^t_{1:K} \in \mathbb{R}^{K\times H\times W}$, $f^t_{(K+1):2K} \in \mathbb{R}^{K\times H\times W}$, and $f^t_{(2K+1):C} \in \mathbb{R}^{(C-2K)\times H\times W}$. Then we shifts the channels along the temporal dimension following [7]:

$$f^t_{agg} = \mathcal{C}(f^{t-1}_{1:K}, f^{t+1}_{(K+1):2K}, f^t_{(2K+1):C}), \tag{1}$$

where $\mathcal{C}(\cdot)$ is the concatenation operation. According to [7], the shift operation is only performed at the residual layer to preserve the spatial feature learning capability of the siamese tracker. Note that the aggregated feature f^t_{agg} has the same shape with f^t, so we can insert this module into the backbone directly, without the need to change other part of the network. What's more, this operation only needs to do data movement, so it is computationally free and can be trained end-to-end.

Feature modulation module After getting the template feature f_z and the search feature pyramid $F_x = \{f^i_x\}_{i=1:5}$, they are modulated to generate target-specific features. Specifically, The modulation vector f_{avg} is generated from f_z using global average pooling, which carries the target-specific appearance information. The modulated feature pyramid $F_{mod} = \{f^i_{mod}\}_{i=1:5}$ is generated as follows:

$$f^i_{mod} = \mathcal{M}(f_{avg}, f^i_x), \tag{2}$$

where $\mathcal{M}(\cdot)$ is the depth-wise correlation [3]. Each modulated feature map is fed into two sibling fully-connected layers—a box-regression layer with channel dimension $4k$, and a box classification layer with channel dimension $2k$, where k is the number of maximum possible proposals for each location. The object/background criterion and bounding box regression are defined with respect to a set of anchors. Following [13], we assign anchors with the same scale to each of the different pyramid levels. For detail information of the anchor setting, please refer to [13]. We use the top-N ranked proposal regions for the refine stage.

2.2. Adversarial dropout module

The refine stage aims to select the target from candidate proposals. Features of these candidate proposals are cropped from the search feature pyramid F_x using RoIAlign [14], and then fused with the target feature f_z:

$$\mathcal{X} = \mathcal{R}(b, F_x) \odot f_z, \tag{3}$$

where $\mathcal{R}$ represents the RoIAlign, $\odot$ represents the element-wise multiplication, b represents an RoI in candidate proposals and $\mathcal{X}$ represents the fused feature of b.

Adversarial dropout After the feature fusion, we use adversarial dropout [10, 15] to increase the discriminative ability of $\mathcal{X}$. We first predict the adversarial dropout mask based on divergence maximum. The mask is applied to $\mathcal{X}$ to get the adversarially dropped features. Then, we aim to minimize the divergence between the randomly dropped features and the adversarially dropped features. Specifically, let h^{cls} and h^{reg} denote the classification layer and the regression layer in stage 2, respectively. The adversarial dropout mask is calculated as follows according to [15]:

$$\begin{aligned} &\mathbf{m}^{adv} = \arg\max_{\mathbf{m}} D[h^{cls}(\mathcal{X} \odot \mathbf{m}^s), h^{cls}(\mathcal{X} \odot \mathbf{m})] \\ &where \; ||\mathbf{m}^s - \mathbf{m}|| \leq \delta_e L, \end{aligned} \tag{4}$$

where L represents the dimension of $\mathbf{m} \in \mathbb{R}^L$, $\mathbf{m}^s$ represents the random mask and $\mathbf{m}^{adv}$ represents the adversarial mask. δ_e is a hyper parameter to control the perturbation magnitude with respect to $\mathbf{m}^s$ [15]. $D[p, p'] \geq 0$ measures the divergence between two distributions p and p'.

To calculate $\mathbf{m}^{adv}$, [10] optimizes a 0/1 knapsack problem with appropriate relaxations in the process. Please refer to [10] for detail information. After generating $\mathbf{m}^{adv}$, we then aim to minimize the divergence between two predicted distribution regarding to $\mathcal{X}$: one with a random dropout mask $\mathbf{m}^s$ and another with an adversarial dropout mask $\mathbf{m}^{adv}$ [15].

$$\mathcal{L}_{adv} = \mathbb{E}[D_{KL}[h^{cls}(\mathcal{X} \odot \mathbf{m}^s)||h^{cls}(\mathcal{X} \odot \mathbf{m}^{adv}))]], \tag{5}$$

where D_{KL} is the Kullback-Leibler divergence.

Finally, for each RoI, the classification layer produces softmax probability estimates over two classes (foreground or background) and the regression layer outputs four real-valued numbers for the foreground class. These four values encode the refined bounding-box position for the RoI. The loss of SiamTFA is:

$$\mathcal{L} = \mathcal{L}^{stage1}_{cls} + \mathcal{L}^{stage2}_{cls} + \mathcal{L}^{stage1}_{reg} + \mathcal{L}^{stage2}_{reg} + \lambda\mathcal{L}_{adv},$$

where λ is a hyper-parameter to balance the adversarial loss and the classification/regression loss. $\mathcal{L}^{\cdot}_{cls}$ is the cross entropy loss and $\mathcal{L}^{\cdot}_{reg}$ is the standard smooth $L1$ loss for regression. During testing, the RoI with the top classification score is selected as the predicted target.

3. EXPERIMENTS

In this section, we first present the implementation details. Then we evaluate out method on GOT-10K [16] testing set and the UAV20L [17] dataset.

Table 1. Performance of our algorithm with different components on GOT-10k test set.

Temporal aggregation	Adversarial dropout	AO	$SR_{0.50}$	$SR_{0.75}$
		0.542	0.607	0.456
✓		0.561	0.645	0.480
✓	✓	0.577	0.662	0.509

Table 2. Comparing the results of our approach against other approaches over the GOT-10k test set. The trackers are ranked by their average overlap (AO) scores.

Method	AO	$SR_{0.50}$	$SR_{0.75}$
Ours	**0.577**^{1}	**0.662**^{1}	**0.509**^{1}
SiamMask	0.459	0.560	0.205
SiamFCv2	0.374	0.404	0.144
SiamFC	0.348	0.353	0.098
GOTURN	0.347	0.375	0.124
CCOT	0.325	0.328	0.107
ECO	0.316	0.309	0.111
CF2	0.315	0.297	0.088
MDNet	0.299	0.303	0.099

3.1. Implementation details

The proposed network is trained on the training set of GOT-10k [16] and the backbone is pretrained on ImageNet. We apply stochastic gradient descent with momentum of 0.9 and set the weight decay to 0.0005. The learning rate is decreased from 10^{-2} to 10^{-4}. The batch size is set to 2 and the network is trained for 90000 iterations. Our tracker is implemented in Python, using PyTorch.

3.2. Evaluation on GOT-10k dataset

In this subsection, we evaluate our method on GOT-10k [16] dataset. GOT-10k is a recent large-scale high-diversity dataset consisting of over 10,000 video sequences with targets annotated by axis-aligned bounding boxes. The GOT-10k testing set includes 180 sequences with 84 different object classes and 32 motion patterns. As performance measure, we use the average overlap (AO) scores and success rate (SR) as proposed in [16]. The AO denotes the average of overlaps between all groundtruth and estimated bounding boxes, while the SR measures the percentage of successfully tracked frames where the overlaps exceed 0.5/0.75.

Ablation Studies From Table 1 (the 1^{st} and 2^{nd} row), we see that the AO performance increases by 3.1% by adding the temporal aggregation module. This is because the temporal aggregation module improves the per-frame features by aggregating temporal information from adjacent frames. From Table 1 (the 2^{nd} and 3^{rd} row), we see that with the adversarial dropout module, the AO increases by 2.9%. This is because the adversarial dropout module improves the discrimination power of our siamese tracking network.

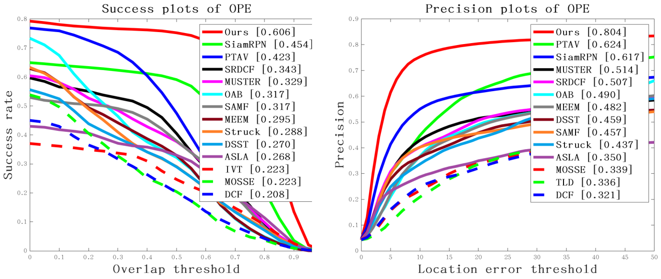

Fig. 3. Success and precision plots on UAV20L dataset.

Overall Performance We compare our proposed method with 8 trackers, including state-of-the-arts. The success plot of the evaluated trackers is shown in Table 2. Compared to other listed approaches, our approach achieves a superior AO of 0.577. Compared with SiamMask, our tracker aims to make full use of the temporal information. As a result, our tracker outperforms SiamMask by 11.8% in terms of AO, which highlights the importance of the proposed temporal aggregation module.

3.3. Evaluation on UAV20L dataset

In this subsection, we evaluate our tracker on the UAV20L [17] long term tracking dataset. It contains 20 HD video sequences captured from a low-altitude aerial perspective with average sequence length of 2934 frames. In this experiment, all trackers are compared using two measures: precision and success. Precision is measured as the distance between the centers of the predicted bounding box and the corresponding ground truth bounding box. Success is measured as the intersection over union of pixels in predicted bounding box and those in ground truth bounding box. In Fig. 3, we can find that the proposed algorithm achieves better tracking performance compared with some representative trackers. In the success plot, our tracker obtains an AUC score of 0.606. In the precision plot, the proposed algorithm obtains a score of 0.804. It shows that our tracker surpass other state-of-the-art algorithms, such as SiamRPN [18] and PTAV [19]. This demonstrates the effectiveness of our tracker in long-term tracking scenario.

4. CONCLUSION

In this paper, we introduce a novel siamese architecture for visual object tracking. Specifically, our proposed algorithm contains two main modules, *i.e.* temporal aggregation module and adversarial dropout module. The temporal aggregation module improves the per-frame features by aggregating features of adjacent frames. The adversarial dropout module improves the discrimination power of the siamese tracking network. Extensive experimental results show that the proposed algorithm performs favorably against the state-of-the-art algorithms.

5. REFERENCES

[1] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng, "A twofold siamese network for real-time object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4834–4843.

[2] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu, "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 101–117.

[3] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[4] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei, "Flow-guided feature aggregation for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 408–417.

[5] Kai Xu, Longyin Wen, Guorong Li, Liefeng Bo, and Qingming Huang, "Spatiotemporal cnn for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1379–1388.

[6] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 548–557.

[7] Ji Lin, Chuang Gan, and Song Han, "Tsm: Temporal shift module for efficient video understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7083–7093.

[8] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1420–1429.

[9] Susanna Gladh, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg, "Deep motion features for visual tracking," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 1243–1248.

[10] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon, "Adversarial dropout for supervised and semi-supervised learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[15] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong, "Drop to adapt: Learning discriminative features for unsupervised domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 91–100.

[16] Lianghua Huang, Xin Zhao, and Kaiqi Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2019.

[17] Matthias Mueller, Neil Smith, and Bernard Ghanem, "A benchmark and simulator for uav tracking," in *European conference on computer vision*. Springer, 2016, pp. 445–461.

[18] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.

[19] Heng Fan and Haibin Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5486–5494.