Switch and Refine: A Long-Term Tracking and Segmentation Framework

Xiang Xu, Jian Zhao, *Senior Member, IEEE,* Jianmin Wu, and Furao Shen, *Member, IEEE*

***Abstract*—In long-term video object tracking (VOT) tasks, most long-term trackers are modified from short-term trackers, which contain more and more machine learning modules to improve their performance. However, we empirically find that more modules do not necessarily lead to better results. In this paper, we make the long-term tracking framework simple by carefully selecting the cutting-edge trackers. Specifically, we propose a new long-term VOT framework that combines the benefits of two mainstream short-term tracking pipelines, i.e., the discriminative online tracker and the one-shot Siamese tracker, with a global re-detector awakened when the target is lost. Such a framework fully exploits existing advanced works from three complementary perspectives. Experimental results show that by exploiting the capabilities of existing methods instead of designing new neural networks, we can still achieve remarkable results on seven long-term VOT datasets. By introducing a continuous adjustable speed control parameter, our tracker reaches 20+FPS with only a small performance loss. The refine module not only improves the bounding box estimations but also outputs segmentation masks, so that our framework can handle the video object segmentation (VOS) tasks by using only VOT trackers. We obtain a trade-off between time and accuracy on two representative VOS datasets by only using bounding boxes as the initial input.**

***Index Terms*—visual object tracking, long-term tracking, visual object segmentation**

I. Introduction

VIDEO object tracking (VOT) has great potential in real-world applications like robotics, autonomous driving, and video surveillance. Until now, most of the research attention has been on short-term tasks, in which targets are always in view and trackers are asked to report the targets' location in every frame. However, a more realistic setting in a long-term task is that trackers can also determine the target disappearance and reappearance.

The majority of the short-term trackers can be divided into two categories: the online learning method and the one-shot learning method. Representative online learning methods [1]–[5] update their templates in the tracking process to adapt to the changes of scale and appearance. They are robust and have set several state-of-the-art (SOTA) results on well-recognized datasets. In contrast, the test phase is totally offline in one-shot learning methods [6]–[11]. Such methods enjoy faster speed with better scale estimation.

It is a common practice to build a long-term tracker composing short-term ones. Existing long-term tracking frameworks [12], [13], especially the VOT challenge entries [14]–[16], use one short-term tracker as a local tracker and another one as a verifier. However, those research does not address the reason that specific short-term trackers are chosen to be the corresponding components in their long-term tracking frameworks. Some are simply carried over from previous research. Instead, we carefully select the cutting-edge short-term trackers by using both the online learning methods and the one-shot learning methods and making them complementary.

In order to bridge the gap between long-term and short-term tracking, previous works have introduced different modules into the long-term tracking framework. Object detectors are used as the global re-detectors in [12]. A target candidate association network [17] keeps tracks of distractor objects to continue tracking the real target. People have also investigated the usage of graph convolutional networks to promote robustness [18]. Nowadays, a common practice to improve the long-term tracking performance is to design and add new neural networks. However, we empirically find that more modules do not necessarily lead to better results. Moreover, the capabilities of previous trackers have not been fully exploited. We will show that combining existing methods can still achieve performances comparable to current SOTA trackers.

In this paper, we propose a **Switch and Refine tracking Framework (SRF)** for long-term video object tracking tasks. Instead of doing addition as previous works, we borrow an idea from ensemble learning that "many could be better than all" [19], i.e., doing subtraction and making the entire framework simpler. Our framework can fully exploit the capabilities of the current advanced works from three complementary perspectives, i.e., discriminative online trackers, one-shot learning offline trackers, and global re-detectors. Each frame is checked with templates at least twice, taking advantage of both the first frame and the latest tracking results. Our switch mechanism is quite simple, since it only takes the confidence score output of the local and the global part into consideration, with no additional verifiers. We also introduce a Speed Control Parameter (SCP), which decides the specific confidence score threshold. To the best of our knowledge, SRF is the first long-term tracker whose speed is continuously adjustable. Evaluation on various long-term tracking datasets with different criteria [14], [15], [20]–[24] has shown the potential of our tracker.

The tracking task now has a trend toward estimating pixel-level tracking results [16]. Such a task can be seen as the

This work is supported by the State Grid Corporation of China under project No. 520950200009. Corresponding authors: Furao Shen, Jian Zhao.

X. Xu and F. Shen are with the State Key Laboratory for Novel Software Technology and School of Artificial Intelligence, Nanjing University, Nanjing, Jiangsu 210023, China. (e-mail: xux@smail.nju.edu.cn; frshen@nju.edu.cn)

J. Wu is with the State Grid Shanghai Maintenance Company, Putuo, Shanghai 200063, China. (e-mail: wujm0987@aliyun.com)

J. Zhao is with School of Electronic Science and Engineering, Nanjing University, Nanjing, Jiangsu 210023, China. (e-mail: jianzhao@nju.edu.cn)

existing semi-supervised video object segmentation (VOS) problem, with a difference in the evaluation criteria. To cater to this trend, our tracking framework also estimates a segmentation mask on every frame for each target, which is generated by the refine module.Our framework is among the few that can accomplish both VOS and long-term VOT tasks. Compare to other works that also take bounding boxes instead of masks in initialization [11], [25], SRF reaches a trade-off between speed and accuracy.

In summary, the contributions of this paper are as follows:

- We prove that by only combining existing methods instead of designing new networks, we can still obtain SOTA performances. Our SRF fully exploits the capabilities of current advanced works from three complementary perspectives.
- We borrow an idea from ensemble learning that "many could be better than all". Experimental results prove that doing subtraction instead of adding modules also achieves remarkable precision in long-term tracking tasks.
- We introduce a parameter SCP to interfere in the switch mechanism between the local tracker and the global redetector. To the best of our knowledge, SRF is the first long-term tracker whose speed is continuously adjustable.
- This work is one of the current few studies on combining VOS tasks and long-term VOT tasks. We show that semi-supervised VOS tasks can be accomplished by taking full advantage of existing VOT trackers.

II. Related Work

A. Visual Object Tracking (VOT)

In a VOT task, a tracker tracks an object through a video given the first-frame bounding box of the target object. Based on the video lengths, VOT tasks are usually divided into two types, short-term and long-term. Targets are supposed to exist in every frame of a video in short-term scenarios. However, trackers have to determine the target disappearance and redetect the target once it appears in a long-term task, which is a more difficult and realistic setting.

1) Short-term tracking: A large number of long-term trackers [12], [13], [26]–[28] are based on short-term ones, so here we provide a brief introduction of short-term trackers.

Most short-term trackers follow Siamese designs which are composed of two branches: a template branch and a search branch. The template branch produces deep features or convolutional filters based on the first frame ground truth or tracking results of previous frames, which are then cross-correlated or convoluted with the current-frame feature map generated by the search branch. Computed heatmaps are then passed through the classification branch and regression branch to generate the final bounding box estimation.

SiamRPN [6] and ATOM [1] are regarded as representative trackers of offline and online methods, respectively. Generally speaking, offline methods, which are often named after *Siam*- have faster test speed and better scale estimation, while online methods produce more precise and robust classification results by making full use of previous tracking outputs.

Enormous works try to boost the performance. Besides widely adopted backbone modification [7], [29], [30], recent efforts include better meta-learning [2], [4], better anchor-free method [8], [30], utilizing pixel-level information [10], [11] and temporal information [5], [31], [32], using multi-level backbone features [4], [7], doing cross-correlation differently [33], [34] and using existing object detection network [35]. Recent methods [36] combine multiple backbone networks. The usage of transformer [37], [38] and different loss functions [39] has also been explored.

2) Long-term tracking: Long-term tracking is first defined in [40], which introduces a framework that tracks and detects the target simultaneously with a learning-from-error mechanism. Following the widely used short-term and long-term interaction mechanism [41], many long-term tracking frameworks nowadays are modified from short-term trackers or use short-term trackers as part of themselves. This methodology requires frameworks to handle the missing target. RLT-DiMP [27] applies random search with spatio-temporal constraints to prevent sudden detection at a distance. For robustness, it augments various unseen backgrounds for more discriminative feature learning and estimates location from multiple images with random erasing. Methods like DeepMTA [42] and mlpLT [43] run the short-term and long-term trackers simutaneously in each frame, consuming much more computing power. In contrast, we aim to design a simple mechanism to switch between the two trackers in this paper, which runs much faster.

Location estimation with a confidence score from a local tracker might not be suitable for direct usage in a long-term scenario. Above metioned works [12], [13], [26] add a verification network to identify the target from candidates generated by the short-term tracker. In implementation, another short-term tracker is usually used as the verifier. Like a sliding window, SPLT [13] applies a SiamRPN [6] based local tracker with a verifier on possible local regions to find the target when it is missing. Its pre-trained skimming module selects such possible regions. A similar strategy is used in LTMU [12], while its possible regions are decided by a faster R-CNN detector [44]. LTMU also introduces a meta-updater that takes historic geometric and appearance cues as input to guide the local tracker and verifier update.

Another long-term tracking pipeline [25], [45] is solely based on a two-stage object detection framework [44], in which a Siamese architecture is embedded in its RPN and/or RCNN. These methods make no assumptions on the temporal consistency of the targets' positions and scales to evade cumulative errors. In Siam R-CNN [25], dynamic programming is further applied to select the best tracklet, which has shown effects on distinguishing distractors. A similar multi-trajectory analysis is used in DeepMTA [42]. However, using the whole picture as the input of each frame is relatively slow.

There also exist keypoint-based trackers. ALIEN [46] takes multiple instances of the same features in different conditions to build a target/background discriminative classifer, which is useful for the template update problem. MUSTer [47] maintains the feature keypoints of both the target and the background in a keypoint database, and finds the best result by both tracking the keypoints between two consecutive frames and

matching the keypoints in the database. Multi-task learning and metric learning also show potential in dealing with drastic changes and distractors [48]. However, such keypoint-based methods adopt a variety of descriptors for keypoint feature representation and apply classical machine learning algorithms for classification, which restrict their tracking abilities. Instead, our framework uses a deep learning approach.

Some methods use hand-crafted or color features instead of deep ones. Multiple candidates are produced using particle filters in [49], which are then fed into the re-detection module followed by a complicated switch mechanism. The input of our re-detector is the entire frame, not only the part of the candidates. Our switch mechanism is much simpler, only taking the confidence score output of the local and the global part into consideration. Classical methods like Kalman filters are applied in CALT [50] for producing candidates, among which the best was chosen by counting contour pixels and estimating reliability using a verifier. In contrast, our method uses no verifier but a refiner to optimize the location estimation in a totally deep-learning way. e-TLD [51] focuses on long-term tracking using event camera, while our framework is designed for standard image stream from an ordinary camera. ASNet [52] is designed for multi-drone single object tracking, while our method is for general cases, including the single object tracking from unmanned aerial vehicle (UAV).

B. Video Object Segmentation (VOS)

VOS tasks can be seen as extensions of VOT tasks where segmentation masks of targets in every frame are required to report. This requires trackers to estimate not only precise object locations but also the contours of the objects, which is more difficult. Classical tools like particle filters and color histogram similarity are used to construct trackers in [53]. They are applied from experience and lack the ability of learning features automatically. Some methods only identify object classes that appear in the training set without the ground-truth mask annotations during testing [54]. However, we only consider the so-called *semi-supervised* scenario, where the trackers track the targets based on the annotated masks in the first frame. Such a task challenges trackers to track unseen objects and is relatively similar to a VOT task. Most VOS methods [55]–[59] use the annotated masks in the first frame for initialization and making trade-offs between real-time speed and good performance. Whereas in this paper, we try to tackle this problem of producing mask results by only using the given template bounding box. Very few efforts have explored this problem. Those few efforts include LWL [60], SiamMask [11] and SiamR-CNN [25]. LWL and SiamMask are designed and experimented on short-term videos. SiamR-CNN is the only tracker that can perform long-term tracking except for VOS tasks, but its mask output is based on its bounding box estimation. In other words, it outputs bounding boxes and masks subsequently. In contrast, our method refines bounding box prediction and mask prediction at the same time.

Nowadays researches on both tracking and segmentation tasks have a trend towards more realistic settings. For example, with the wide usage of drones, both the datasets and solutions for detection, tracking and counting dense crowds from the air have been developed [61]. Pixel-wise human annotations are expensive in real-world applications. SSVOS [62] is designed to achieve video object segmentation with scribble-level supervision. A more common situation is that videos are only associated with textual descriptions. SPFTN [63] takes advantage of both multi-task and self-paced learning to localize and segment objects in such weakly labeled videos. Some methods [64] use subsequent frames to filter out indistinguishable proposals. Compared to those works, we want to build an online tracker by only using information from the past observations.

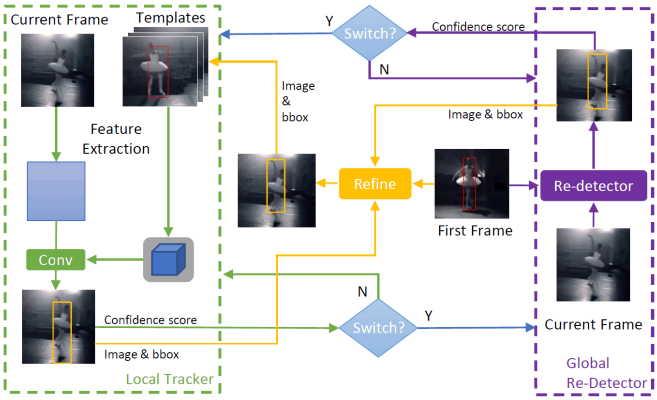


Fig. 1. An overview of our framework. SRF consists of three parts: a local tracker, a global re-detector, and a refine module. When the framework has enough confidence that the target is in the view, the local tracker is used. The re-detector is awakened when the object is supposed to be lost. The switch mechanism decides when to go back to the local tracker. Results from both the local tracker and the global re-detector are passed through the refine module to produce the final bounding box (and mask if needed).

III. Method Description

As shown in Fig. 1, our framework consists of three parts, a local tracker, a global re-detector, and a refine module. The framework assigns the local tracker to track initially and asks the global re-detector for help when the local tracker misses the target. Without additional verifiers or temporal modules, the framework switches between the local tracker and the global one by solely considering their confidence scores. Before going deep into specific parts of our framework, we first introduce the design thinking behind it.

A. The Design Thinking: Many Could be Better Than All

Recent long-term tracking mechanisms are becoming more bloated with lower speed. To obtain better performance, lots of components are added. For instance, four trackers (ATOM, SiamMask, SiamRPN, RT-MDNet), an object detector (Faster R-CNN) and an LSTM-based meta-updater (including MetricNet [65]) are included in LTMU. However, we empirically find that such a combination may not monotonically boost the performance, as shown in Table I. Both the MetaUpdater and the MetricNet can interfere in the template update mechanism. They may choose the wrong frames, and such bad choices would accumulate, which we believe is the reason that leads to poor performance when these two components are used.

TABLE I
"MANY COULD BE BETTER THAN ALL"

	Components	F-score
A	SuperDiMP	0.664
B	A+SiamR-CNN	0.695
C	B+AlphaRefine	**0.705**
D	C+MetricNet	0.692
E	D+MetaUpdater	0.695

To improve the performance, previous works usually do the addition, i.e., adding more components. However, more components mean slower speed with heavier computational complexity, which does not necessarily lead to better performance. In contrast, we want to do some subtraction, which we learn from a phenomenon in ensemble learning called "many could be better than all" [19]. In other words, we want to design a long-term tracking framework with fewer components but achieve SOTA performances. We also want to know whether designing a totally new neural network is the only way to boost the evaluation metrics. We try to achieve high precision and recall by exploiting the capability of existing trackers.

In previous works, there is no mention of how the respective short-term trackers are chosen. As mentioned in Section II, the latest deep trackers can be divided into two classes. Discriminative online trackers update their templates during the testing phase and achieve higher robustness. Offline trackers that are usually named after Siam- are good at scale estimation and run fast. We thus try to combine these two kinds of trackers, utilizing the advantages of both. For discriminative online trackers, methods derived from DiMP [2] are current SOTA on short-term video clips, and we simply choose the latest one, TrDiMP [37], as our local tracker. The fast speed of the offline trackers makes them suitable as a refiner. We use the Siamese-designed AlphaRefine [33] as our refine module.

A global re-detection mechanism is ubiquitous in the previous long-term tracking framework [12], [13]. Widely used sliding-window approach seems to be brute, and an object detection model only recognizes its pre-defined categories, for instance, the 80 categories in COCO [66]. GlobalTrack [45] and SiamR-CNN [25] are two models that directly embed Siamese architecture into the detection pipelines. A very natural idea is applying these methods as a global re-detector in a long-term tracking framework. However, it is challenging to combine the tracklet dynamic programming algorithm introduced in the original SiamR-CNN with other modules, while GlobalTrack produces unsatisfactory results. To compromise, we use a simplified SiamR-CNN as our global re-detector, which is easier to use with faster speed and less accuracy loss (see Table VIII) compared to the original one, but still has better performance than the GlobalTrack.

B. Local Tracker

The local tracker tracks the target based on movement continuity when the framework has enough confidence that the target is in view, for example, in the beginning. The local tracker outputs the top-1 bounding box prediction $\mathrm{bbox}_{\mathrm{st}}(\mathbf{x})$ with its confidence score $\mathrm{score}_{\mathrm{st}}(\mathbf{x})$. For simplicity, we use $\mathbf{x}$ to denote both the search frame and the search patch, a partial region of the search frame usually centering at the target position in the last frame. Focusing on a small area may boost the speed and improve the accuracy of the local tracker.

Theoretically, any existing short-term trackers can be used as a local tracker. In our framework, we utilize TrDiMP [37] as our local tracker. It takes advantage of both the solid discriminative tracking pipeline and the transformer's strength in capturing temporal contexts.

In the transformer, the encoder takes the features of the template patch $\Psi(\mathbf{z})$ as the input, where Ψ is the backbone network and $\mathbf{z}$ represents the template patches. The encoded information $\mathrm{Enc}(\cdot)$ from the encoder is fed into the decoder, along with features from the search patch $\Psi(\mathbf{x})$. Following the discriminative tracking pipeline, the encoder also produces a convolution kernel or a tracking model f. The kernel f is optimized under a ridge regression via a meta-learner as:

$$\min_{f} ||f * \mathrm{Dec}(\mathrm{Enc}(\Psi(\mathbf{z})), \Psi(\mathbf{x})) - \mathbf{y}||_2^2 + \lambda ||f||_2^2. \quad (1)$$

The local tracker minimizes the average of the objective function in (1) over the whole training set. In real tasks, the accurate location of a target is hard to define using only a single coordinate. Instead, a heatmap derived from a Gaussian distribution turns the coordinate into a *soft* label, providing richer information and making the learning process easier. In (1), $\mathbf{y}$ is the Gaussian-shaped ground-truth label of template patch $\mathbf{z}$, and λ is the regularization term. Such a meta-learning way captures the distinctive feature of each video. The kernel is further convoluted with the decoded features $\mathrm{Dec}(\cdot,\cdot)$ to produce the final feature map $\mathbf{r}$:

$$\mathbf{r} = f * \mathrm{Dec}(\mathrm{Enc}(\Psi(\mathbf{z})), \Psi(\mathbf{x})). \quad (2)$$

The target is supposed to be with the highest score $\mathrm{score}_{\mathrm{st}}$ on the 2-dimensional feature map $\mathbf{r}$, i.e.,

$$\mathrm{score}_{\mathrm{st}}(\mathbf{x}) = \max_{i,j} \mathbf{r}_{i,j}. \quad (3)$$

Following DiMP [2], the bounding box estimation $\mathrm{bbox}_{\mathrm{st}}$ is produced by an IoUNet [1].

Transformer is a hot topic in recent tracking literature [37], [38]. Some trackers [38] use it to model the sequence relationship, just as in the NLP domain. While in our local tracker, the transformer is regarded as a substitution of the widely-used cross-correlation operation. Please refer to [37] for the details of the encoder and the decoder.

C. Global Re-detector

When a target is supposed to be out of view, or we do not have enough confidence about the candidates produced by the short-term tracker, we have to search the object in the entire frame. An object detector is suitable for such remedy, with a vital difference that our target is defined by the first-frame ground truth, not the predefined object classes. Existing frameworks combine a detector and a tracker as a whole to use first-frame annotations [12]. However, our framework directly

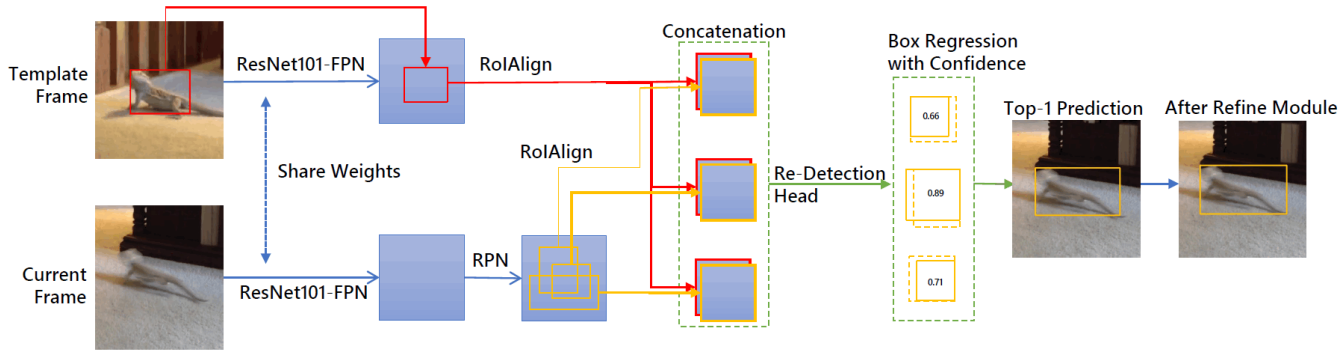


Fig. 2. The structure of our global re-detector. We utilized a simplified SiamR-CNN, which shares a similar structure with an object detector. The RoI-aligned template features and the RoI-aligned region proposal features are concatenated by a 1×1 convolution to reduce the number of channels down by half. The re-detection head is applied on the fused feature for box regression, then simply pick the bounding box with the highest confidence score as the result. Comparisons between the results before and after the refiner are shown in Fig. 3.

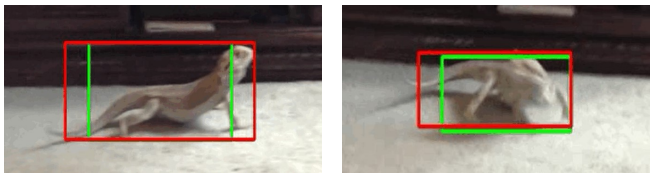


Fig. 3. Effectiveness of the refine module. Results before and after the refine module are in green and red respectively. Red boxes bound the target more precisely.

uses a modified Siamese architecture object detector to find the target of a probably unseen category.

The original SiamR-CNN model [25] utilizes two three-stage cascades [67] for re-detection, one with the first-frame feature followed by the other with the previous-frame feature. Several candidate proposals are assigned with tracklets, and the final prediction is decided on the tracklet score. Although the scheme has effects on distinguishing distractors, we find it is time-consuming and incompatible with the short-term trackers. As a result, our framework uses a simplified version, as shown in Fig. 2.

The search frame $\mathbf{x}$ and the template frame $\mathbf{z}_0$ are separately fed into a backbone network Φ to produce $\Phi(\mathbf{x})$ and $\Phi(\mathbf{z}_0)$. The RPN network produces region proposals on $\Phi(\mathbf{x})$, and deep features extracted by RoI-Align on these proposals are further concatenated with the RoI-Aligned template feature,

$$F(\mathbf{x}) = \mathrm{RoIAlign}(\mathrm{RPN}(\Phi(\mathbf{x}))), \tag{4}$$

$$F(\mathbf{z}_0) = \mathrm{RoIAlign}(\Phi(\mathbf{z}_0)). \tag{5}$$

The re-detection head does box estimation based on the composited feature $[F(\mathbf{x}), F(\mathbf{z}_0)]$, where $[\cdot,\cdot]$ represents the concatenation. 1×1 convolution is used here to make the fused feature have only one channel, so the re-detection head shares the same structure as the detection head in Faster R-CNN. Lastly, our framework simply picks the bounding box $\mathrm{bbox}_{\mathrm{lt}}(\mathbf{x})$ with the top-1 confidence score $\mathrm{score}_{\mathrm{lt}}(\mathbf{x})$ on the long-term heatmap $\mathbf{r}'$as follows,

$$\mathbf{r}' = \mathrm{RedetectionHead}([F(\mathbf{x}), F(\mathbf{z}_0)]), \tag{6}$$

$$\mathrm{score}_{\mathrm{lt}}(\mathbf{x}) = \max_{i,j} \mathbf{r}'_{i,j}. \tag{7}$$

The re-detection head has a regression branch predicting the anchors' location and shape, and a classification branch estimating the confidence score (whether the target is in the anchor or not). During training, the regression branch is learned by a smoothed L_1 loss L_{reg}, and the classification branch uses the classical cross-entropy loss L_{cls}. The two terms are weighted by a balancing parameter, which is set to 10 in implementation,

$$L_{lt} = 10 * L_{reg} + L_{cls}. \tag{8}$$

RoIAlign is applied to facilitate the concatenation of the proposals and references. The global re-detector is awakened only when the short-term tracker cannot find the object well, and such circumstance is often accompanied by changes in object size and aspect ratio. Using RoIAlign, such changes are partly offset, making the template matching easier. We thus conclude that RoIAlign-based concatenation is robust in such a scenario and is better than the popular cross-correlation operation. Note that we use only the first frame as the template in the global re-detector, not constantly updating the templates like the local tracker. We add a subscript '0' to $\mathbf{z}$ to show the difference. Both the concatenation operation and the RoIAlign mechanism contribute to better re-detection performance.

D. Refine Module

Research has shown that pixel-wise annotations produce strong supervision, leading to better scale estimation [11], [12]. In standard tracking datasets, ground truths are given in the format of bounding boxes, but not masks. The framework has to produce masks without any manually annotated ones as prior knowledge. This requires the model to precisely distinguish between target and background (sometimes distractors) given only a few bounding boxes. Pretraining on large-scale image and VOS datasets makes such requirements possible.

Previous long-term tracking frameworks have utilized SiamMask [11] for accurate estimation. Using such an entire tracking framework might be bulky, so we use a lightweight refine module AlphaRefine [33] instead. Our refine module also follows a Siamese design, but it is more concise than the other two parts. The size of the search patch $\mathbf{x}$ is twice the

size of the output from the tracker $\mathrm{bbox}_{\mathrm{st}}(\mathbf{x})$ or the re-detector $\mathrm{bbox}_{\mathrm{lt}}(\mathbf{x})$, much smaller than that of the short-term tracker. After feeding into the backbone network Υ, the search feature $\Upsilon(\mathbf{x})$ and the template feature $\Upsilon(\mathbf{z}_1)$ are used to compute pixel-wise correlation $\star$ to obtain correlation maps $\mathbf{r}''$:

$$\mathbf{r}'' = \Upsilon(\mathbf{x}) \star \Upsilon(\mathbf{z}_0). \tag{9}$$

Different from previous works, our refine module uses pixel-wise correlation for better spatial information retainment. The fused feature $\mathbf{r}''$ is passed through a key-point style bounding box prediction head and an auxiliary mask head to obtain the final estimation $\mathrm{bbox}_{\mathrm{rf}}(\mathbf{x})$. The AlphaRefine applies a concentric search region of two times the target's size, instead of four times as often used, which leads to more precise prediction. The module itself does not output a confidence score. Unlike the off-the-shelf box-to-segmentation network Box2Seg [68], which produces masks based on predicted bounding boxes, our refine module optimizes the bounding box $\mathrm{bbox}_{\mathrm{rf}}(\mathbf{x})$ and the mask $\mathrm{mask}_{\mathrm{rf}}(\mathbf{x})$ simultaneously. The template in our refiner is always the first-frame ground truth $\mathbf{z}_0$. During training, mean squared error L_{box} and binary cross-entropy loss L_{mask} are used for the box output and the mask output respectively. The total loss of the refine module L_{rf} is the weighted sum of the two losses:

$$L_{rf} = L_{box} + 1000 * L_{mask}. \tag{10}$$

E. Time Controllable Switch Mechanism

Given the above three components of our framework, it is of great importance to decide when to use each. We use a flag st_flag to mark the state. In the beginning, st_flag is true by default, and the short-term tracker $\mathrm{ST}()$ tracks the target in the area around its (predicted) location in the last frame. The global re-detector $\mathrm{LT}()$ is assigned (set st_flag to false) to search on the current frame *again* entirely if the short-term tracker cannot find the target or feel uncertain ($score_{st} < \eta$) about its prediction. Run the re-detector until the confidence score of its prediction is above threshold θ for successive K frames when we suppose the target is in the view and is *not detected by chance*. SRF updates the latest prediction into the short-term tracker template and passes information like current position and target scale to the tracker. The framework sets st_flag to true and switches back to the short-term tracker again from the next frame. In each frame, bounding box estimation from either the short-term tracker or the global re-detector is fed into the refine module to get the final output. Given that each frame is checked twice (in the tracker or the re-detector and in the refiner), we name it a double-check mechanism. The processing flow of our method is summarized in Algorithm 1. By default, we set $K = 2$ and $\theta = 0.7$.

Speed is one of the core concerns in real applications. As shown in (11), we introduce a parameter SCP in the decision of the confidence score threshold η of the short-term tracker, so as to control the switch mechanism between the local tracker and the global re-detector. The mechanism is built on the fact that the local tracker runs faster than the global re-detector on one frame. The SCP is the percentile of the descendingly sorted short-term tracker's confidence score of all the untracked frames in pre-existing tracking datasets. SRF produces the best results when the parameter equals to 0, the default value, which means that all the untracked frames in the short-term tracker are fed into the long-term one. With the parameter moving towards 1, η becomes smaller, and more frames would directly accept the results from the short-term tracker, and the re-detector is increasingly likely to be suppressed, thus the overall speed become faster.

Algorithm 1 Processing Flow of SRF

Input: current frame $\mathbf{x}$
Output: refined bounding box estimation $bbox_{rf}$

```
if st_flag is True then
    bbox_st, score_st = ST(x)
    if target is not found or score_st < η then
        set st_flag to False
    end if
end if
if st_flag is False then
    bbox_lt, score_lt = LT(x)
    if score_lt > θ then
        update_counter += 1
        if update_counter > K then
            update ST using LT result
            reset update_counter = 0
            set st_flag to True
        end if
    end if
end if
refine the result using either bbox_st or bbox_lt to get bbox_rf
```

$$U = \#\text{untracked frames in dataset using ST},$$
$$\eta = (SCP \times U)_{\text{th}} \text{ conf. score of untracked frames.} \tag{11}$$

IV. Experiment

We evaluate SRF on seven long-term tracking benchmarks and two video object segmentation benchmarks. We tune the hyper-parameters (θ, K) on VOT2019-LT and keep using this set of hyper-parameters on all benchmarks. In other words, we do **not** tune the hyper-parameters on other datasets. The code will be released upon publication.

A. Long-Term Object Tracking Evaluation

Different challenges use different datasets. Videos from VOT-LT and TLP are relatively longer, where targets are sometimes easier to follow by utilizing movement continuity. The local tracker in SRF plays an essential role in such circumstances, and its function is proved by the failure of detect-based trackers like Siam R-CNN. Refiner produces precise results when targets move fast, which frequently happens in UAV and LaSOT. In LaSOTExtSub and the *val* subset of OxUvA, targets do not appear in the train set (including the datasets used for pre-training). We believe that the double-check mechanism for the template (one in tracker/re-detector, another one in refiner) is crucial when tracking in these videos. We introduce the datasets and the performance of SRF on each in the following.

TABLE II
PERFORMANCE EVALUATION ON VOT2018-LT (LTB35) DATASET.

Tracker	F-Score	*Pr*	*Re*
SRF (proposed)	0.713	0.725	0.701
KeepTrack [17]	0.713	0.727	0.703
LTMU [12]	0.690	0.710	0.672
GlobalTrack-RCB [18]	0.681	0.647	0.718
Xuan et al.'s [26]	0.673	0.725	0.628
SuperDiMP [3]	0.671	0.678	0.663
SiamR-CNN [25]	0.668	0.667	0.675
TrDiMP [37]	0.653	0.673	0.635
PrDiMP [3]	0.634	0.646	0.623
SiamRPN++ [7]	0.629	0.649	0.609
SPLT [13]	0.616	0.633	0.600
DeepMTA [42]	0.584	0.544	0.606
CALT [50]	0.410	-	-

TABLE III
PERFORMANCE EVALUATION ON VOT2019-LT (LTB50) DATASET.

Tracker	F-Score	*Pr*	*Re*
SRF (proposed)	0.707	0.717	0.696
KeepTrack [17]	0.709	0.723	0.697
LTMU [12]	0.697	0.721	0.674
LT_DSE [15]	0.695	0.715	0.677
LTMU_B [16]	0.691	0.701	0.681
SiamR-CNN [25]	0.663	0.658	0.669
TrDiMP [37]	0.653	0.673	0.633
SuperDiMP [3]	0.647	0.654	0.641
PrDiMP [3]	0.632	0.641	0.623
SPLT [13]	0.559	0.591	0.530

1) VOT-LT: The 35 sequences in the VOT2018-LT dataset were used in the long-term track of the well-known VOT challenge in 2018 [14]. The dataset was then expanded to 50 sequences, namely VOT2019-LT, which have been used in the challenge since 2019 [15], [16]. Unlike the widely used methodology in detection literature that decides the success and failure on a specific IoU threshold, the VOT initiative designed *tracking* precision (*Pr*), *tracking* recall (*Re*), and *tracking* F-score tailored for the tracking domain, which depend directly on the predicted confidence. To avoid all manually-set thresholds, the final values of these measures are obtained by selecting the confidence threshold that maximizes F-measure, which is tracker-specific optimal. Readers are referred to [69] for details.

Table II shows the results on VOT2018-LT. The trackers are ranked by the tracking F-score, and the top three results are marked in red, blue and green, respectively. SRF achieves an F-score of 0.713, which is the same level as the current best method KeepTrack [17]. Table III compares our results to nine SOTA methods on VOT2019-LT. SRF's results are competitive with the SOTA trackers.

We also provide representative results of our tracker and the two top-ranked methods on the VOT2019-LT dataset. As shown in the top two lines in Fig. 7, SRF loses target less often and produces better scale estimation. However, SRF's

TABLE IV
PERFORMANCE EVALUATION ON LASOT TEST DATASET.

Tracker	Precision	Norm. Precision	AUC
SRF (proposed)	70.8	75.7	67.1
KeepTrack [17]	70.4	77.4	67.2
TransT [38]	69.0	73.8	64.9
SiamR-CNN [25]	68.4	72.2	64.8
TrDiMP [37]	61.4	73.2	63.9
SuperDiMP [3]	65.3	72.2	63.1
PrDiMP [3]	60.8	68.8	59.8
GlobalTrack-RCB [18]	54.5	-	54.0
LTMU [12]	53.5	62.1	53.9
Ocean [30]	52.6	61.0	52.6
GlobalTrack [45]	52.7	59.7	52.1
DeepMTA [42]	47.4	-	52.0

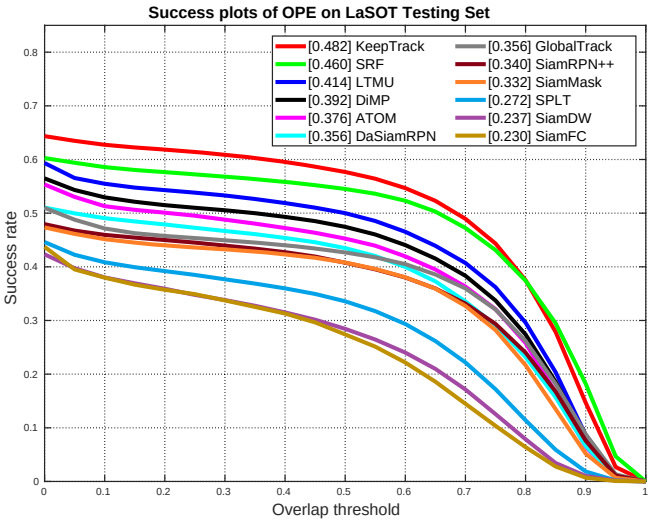


Fig. 4. Success plots of SRF and other 11 methods on LaSOTExtSub dataset.

performance is sometimes limited by its tracker and re-detector when they are disturbed by similar objects, or the target is partially occluded, like the cases in Fig. 9. The reason behind these failures might be that the RoIAlign does not work well when the target is partially occluded by similar object. The RoIAlign itself does not have the ability of distinguishing between objects. It resizes the object so as to ease the template matching. Objects of different sizes may be unified into the same size, for example, a small cat and a big dog in the second row of Fig. 9, which increases the difficulty for distinguishing.

2) LaSOT: LaSOT [20] is one of the largest densely annotated tracking datasets. 280 videos are included in its test dataset, with an average video length of around 2500 frames. LaSOT applies the precision and success rate for quantitative analysis [70]. The precision rate is the percentage of frames whose estimated object center is within a given threshold distance of the ground truth. The distance is normalized by the object scale to compute the normalized precision rate. The success rate is the ratio of predicted bounding boxes whose IoUs with the ground truth are higher than a threshold. By varying the threshold, we can draw a success plot. Trackers

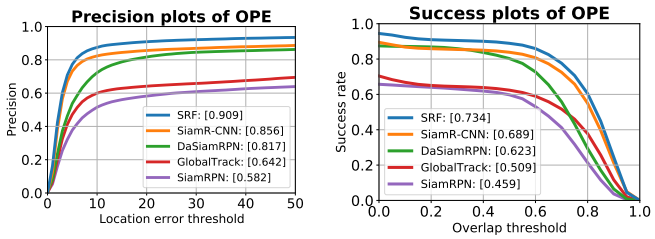


Fig. 5. Precision plots and success plots comparing 5 trackers on UAV20L dataset.

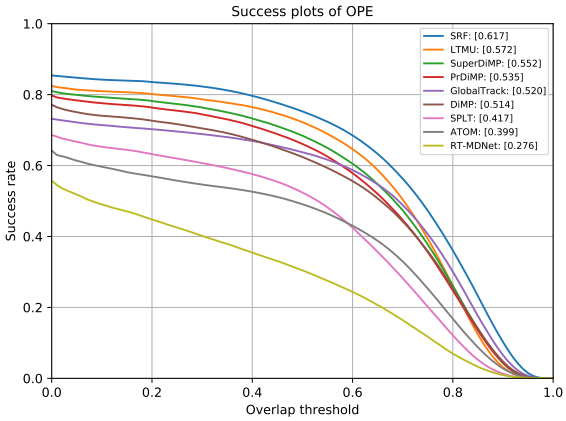


Fig. 6. Success plots of SRF and other 8 trackers on TLP dataset.

are ranked by the Area Under Curve (AUC) of their success plots. As Table IV shows, SRF reaches an AUC of 67.1%, with only a 0.1% gap with the SOTA tracker KeepTrack [17]. The high performance of SRF might contribute to the highest precision among all the methods.

3) LaSotExtSub: LaSOTExtSub [21] is the extended subset of LaSOT, in which 150 sequences contain 15 extra object classes that do not appear in LaSOT's train set. Note that the 15 new classes are selected outside of ImageNet [71]. Such "one-shot" protocol requires long-term trackers to have the ability of tracking previously unseen objects, which is assumed to be more challenging than the LaSOT test set itself.

Bottom two lines in Fig. 7 show how well our SRF tracks previous unseen target even in the circumstances that confusing distractors exist or the target has chaos movement. SRF is better at rediscover after tracking failures than previous methods. We believe that the utilization of movement continuity and the double-check mechanism for the templates help SRF boost its ability.

LaSOTExtSub shares the same evaluation methodology as LaSOT. Fig. 4 is the success rate plot of 12 trackers on the dataset. Though KeepTrack [17] ranks higher when the overall evaluation metrics are used, SRF has an advantage when the overlap threshold is higher than 0.8. SRF might be a better choice when only highly accurate results are needed.

4) UAV20L: UAV20L [24] collects 20 long sequences from low-altitude UAVs with an average length of over 2900 frames. Trackers have to deal with specific tracking nuisances in an aerial environment, such as scale variation and low resolution, from the perspective that other datasets cannot physically or persistently provide.

TABLE V
PERFORMANCE EVALUATION ON OXUVA TEST SET.

Tracker	TPR	TNR	MaxGM
SRF (proposed)	0.819	0.718	0.767
LTMU [12]	0.749	0.754	0.751
Xuan et al.'s [26]	0.625	0.879	0.741
SiamR-CNN [25]	0.701	0.745	0.723
SPLT [13]	0.498	0.776	0.622
GlobalTrack-RCB [18]	0.565	0.680	0.620
GlobalTrack [45]	0.574	0.633	0.603

UAV20L uses precision and success measures to compare tracking methods, similar to LaSOT. Fig. 5 shows the precision and success plots of 5 trackers. Compare to the previous best approaches SiamR-CNN [25] and DaSiamRPN [9], SRF achieves absolute gains of more than 5% in precision and 4.5% gain in success.

5) TLP: TLP [22] is featured by its long average per-sequence duration, making it ideal for studying various problems specific to the long-duration aspect. TLP consists of 50 real-world videos encompassing over 676K frames. Trackers have to distinguish target objects under challenges such as occlusion, fast motion, viewpoint change, scale variations, etc..

As shown in Fig. 6, SRF outperforms the previous top methods by 4.5% absolute gain in success rate, verifying the significant advantage of our approach in tracking long videos.

6) OxUvA: The dataset comprises 166 sequences in its test set, with an average duration of 2.4 minutes along with frequent object disappearance [23]. Its evaluation criteria are different from the widely used "OPE" benchmark. To pursue higher MaxGM, the average of True Positive Rate (TPR) and True Negative Rate (TNR), trackers must be aware of the presence or absence of the target and where it is located. This requires our tracker to have an explicit threshold to decide whether the target is found. Following the common practice, we tune the threshold on the **dev** set and submit the results on the **test** set to the official evaluation server.

Table V is the comparison of our SRF and the other six SOTA trackers. SRF achieves the best TPR ever, which leads to a significant improvement in MaxGM. The MaxGM of 0.767 is 1.6% higher than that of the previous best method, LTMU [12], but there is a big gap in TNR between SRF and Xuan *et al.*'s method [26].

Our experimental results on above datasets show that not adding modules but doing subtraction and keeping the framework simple can achieve remarkable performances in long-term tracking tasks. We also empirically proved that only combining existing methods may still achieve comparable performances as those with newly designed networks.

B. Video Object Segmentation Evaluation

1) DAVIS 2017: Compared to the above tracking datasets, videos in VOS datasets are relatively short but with precise annotated pixel-wise masks. In its evaluation, two complementary measures are introduced. The Jaccard index $\mathcal{J}$ is defined

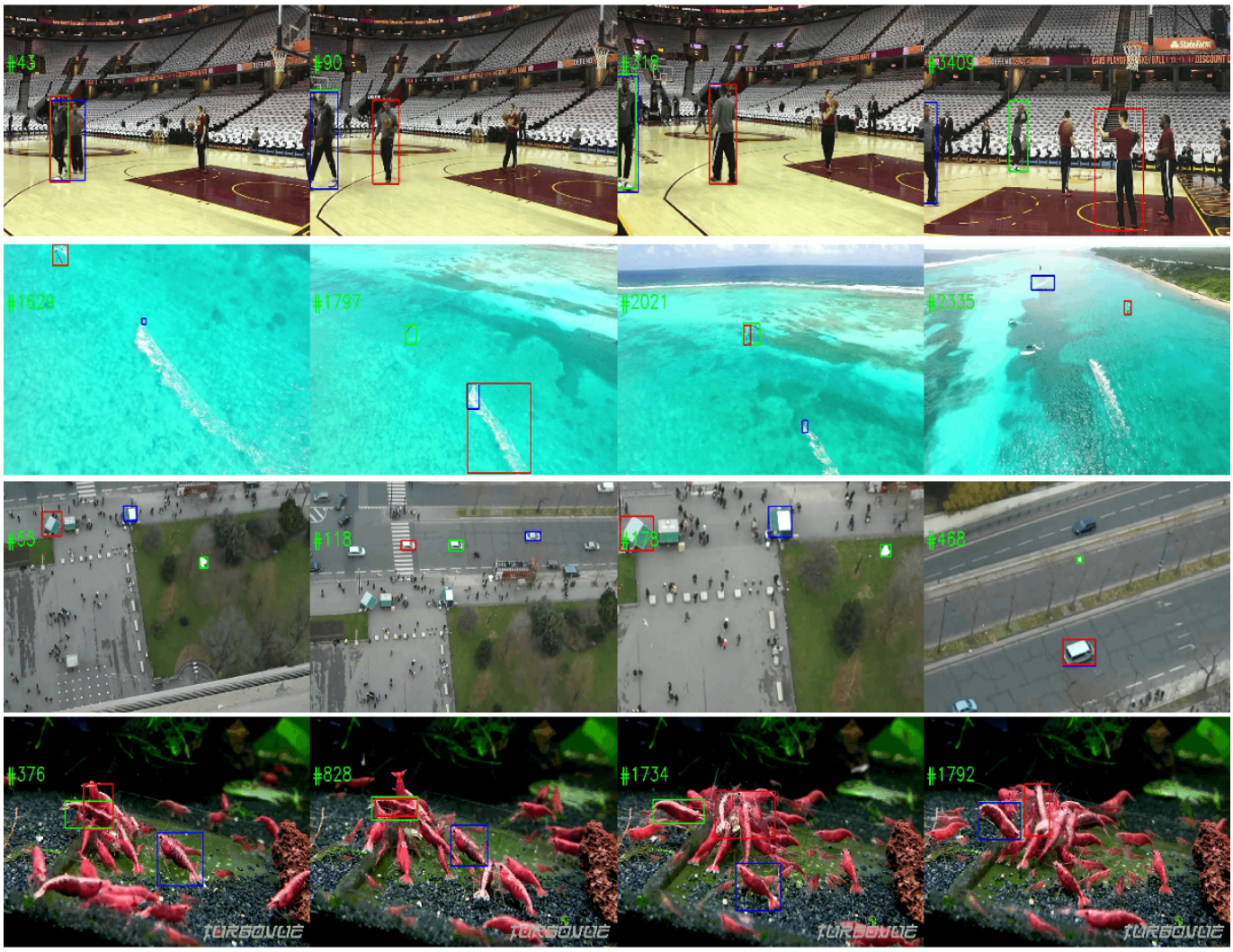


Fig. 7. Visualization and comparisons of representative long-term tracking results of Siam R-CNN (red), SRF (green) and DiMP-based tracker (blue) on VOT2019-LT and LaSOTExtSub. The frame counter is in the upper left corner of each frame (same below). Best viewed in color with zoom-in.
Line 1: warmup, VOT2019LT. The target is the player in the green boxes (SRF's results, close to the groundtruth). The other two trackers are confused with the distractors (other persons wearing similar clothes on the court). Especially in #3409, only SRF tracks the right person.
Line 2: parachute, VOT2019LT. Sometimes the target is easy to follow by utilizing the movement continuity, but Siam R-CNN fails since it detects objects in the entire image (#1629, #2021, #2335). Thanks to the refiner, SRF finds the right target (#1797) and produces more accurate results (#2021).
Line 3: misc-9, LaSOTExtSub. The target is the paper plane. Only SRF tracks the right target in many frames (#55, #178, #468), even after tracking failure (#118). We hold the opinion that SRF's double-check mechanism for the template plays an important role in tracking unseen objects.
Line 4: misc-10, LaSOTExtSub. Many distractors exist, with chaos movement. Even humans are difficult to recognize the targets. This video fully demonstrates SRF's ability to track by movement continuity (#376, #828, #1734, #1792), double-check the template (#1734), and refine (#376).

TABLE VI
PERFORMANCE EVALUATION ON DAVIS2017 VAL DATASET.

Test Input	Train Input	Tracker	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	time
bbox	bbox	SiamR-CNN [25]	0.706	0.661	0.750	0.32
		SRF (proposed)	0.623	0.578	0.667	0.11
		SiamMask [11]	0.558	0.543	0.585	0.02
bbox	mask bbox	LWL [60]	0.706	0.679	0.733	0.17
mask	mask	LWL [60]	0.816	0.791	0.841	0.17
		FRTM-VOS [56]	0.767	-	-	0.05
		TVOS [58]	0.723	0.699	0.747	0.03

TABLE VII
PERFORMANCE EVALUATION ON YOUTUBE-VOS2018 VAL DATASET

Test Input	Train Input	Tracker	Overall	$\mathcal{J}_{seen}$	$\mathcal{J}_{unseen}$
bbox	bbox	SiamR-CNN [25]	0.683	0.699	0.614
		SRF (proposed)	0.623	0.662	0.537
		SiamMask [11]	0.528	0.602	0.451
bbox	mask bbox	LWL [60]	0.702	0.727	0.625
mask	mask	LWL [60]	0.815	0.804	0.764
		FRTM-VOS [56]	0.721	0.723	0.659
		TVOS [58]	0.678	0.671	0.630

Fig. 8. Visualization of representative segmentation results on DAVIS2017. Ground truth is shown in the first frame and predicted results are shown in the following frames. We want to emphasize that SRF uses only bounding boxes but not masks in initialization, and SRF has produced quite satisfactory results. Best viewed in color with zoom-in.

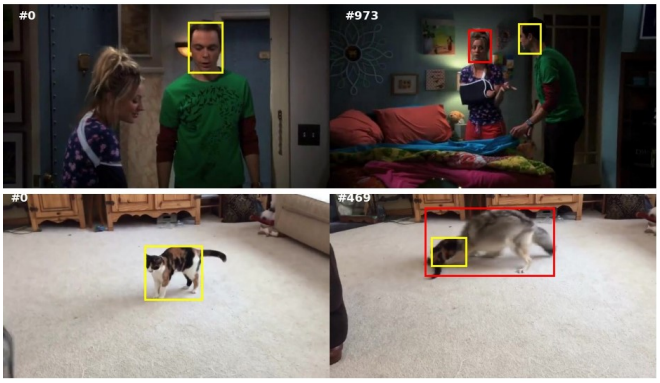


Fig. 9. Failure examples on VOT2019-LT. We compare SRF (red) with ground truth (yellow). The first row is an example of view change. The cat is partially occluded by the dog in the second row. SRF misidentifies the targets in both cases. Best viewed in color with zoom-in.

Fig. 10. Failure examples on DAVIS2017. The first failure may contribute to fast motion. The tracker fails to distinguish between the two competitors. The second failure is caused by the partially occlusion of the targets. Best viewed in color with zoom-in.

as the IoU of the estimated segmentation and the ground-truth mask to evaluate how well the pixels of the two masks match. F-measure $\mathcal{F}$ focuses on contour accuracy, which is robust to slight inconsistency. $\mathcal{J}\&\mathcal{F}$ is the mean of the two metrics. Readers are referred to [54] for details.

Though we have provided some latest works on VOS that take masks as the initial input in Table VI, here we focus on methods that only use the given template boxes as SRF does to make a fair comparison. Compared to SiamR-CNN [25], the most similar tracker to SRF that produces bounding boxes and masks for long-term videos, our framework is about three times faster than it, with a bit of loss of accuracy. To conclude, SRF reaches a balance between speed and accuracy.

It is interesting to note that though LWL [60] also has a version that takes bounding boxes as the initial input in testing, the network design along with its training methods are different from the other three trackers that also use bounding boxes as the test input. LWL first trains its fully utilized version by using masks as its input. Then a bounding box encoder module is plugged in at the beginning of the entire network. This module is trained *alone*, keeping the other network parameters fixed. As a result, the high performance of LWL that takes the bounding box as the test-time initial input is gained from both the pixel-level annotations and box-level annotations. In contrast, the other three works (SiamMask [11], SiamR-CNN [25] and our SRF) are designed for and trained using only box-level annotations. It is a normal phenomenon that more supervised information leads to better results, but we still think that SRF has its value in practice when only bounding box annotations are available.

Fig. 8 provides the qualitative results of SRF. Note that SRF only uses bounding boxes but not masks as the supervised information. Taking this into consideration, we would say that SRF has produced quite satisfactory results. We also provide two failure cases in Fig. 10. In these cases, SRF fails to struggle with fast motion and partially occlusion.

2) Youtube-VOS 2018: Youtube-VOS [72] is a large-scale video object segmentation dataset, whose valid dataset contains 474 Youtube video clips. The valid part includes 26 unique categories that do not appear in its train set to evaluate the generalization ability on unseen objects.

In Youtube-VOS, trackers are evaluated on seen and unseen objects separately, using similar metrics as in DAVIS. The overall metric is the average of the $\mathcal{J}$ metrics on seen objects $\mathcal{J}_{seen}$ and unseen objects $\mathcal{J}_{unseen}$, and $\mathcal{F}$ metrics on them, $\mathcal{F}_{seen}$ and $\mathcal{F}_{unseen}$. Table VII compares the results on the

TABLE VIII
EFFECTIVENESS OF DIFFERENT COMPONENTS ON VOT2019-LT

SiamR-CNN	TrDiMP	Refiner	F-score	Pr	Re
✓(simplified)			0.656	0.652	0.659
✓(complete)			0.663	0.658	0.669
	✓		0.653	0.673	0.633
✓(simplified)		✓	0.661	0.658	0.665
✓(complete)		✓	0.669	0.664	0.675
	✓	✓	0.674	0.692	0.658
✓(simplified)	✓		0.692	0.699	0.685
✓(simplified)	✓	✓	**0.707**	**0.717**	**0.696**

TABLE IX
EVALUATIONS ON DIFFERENT NUMBER OF SUCCESSIVE FRAMES K

K	1	2	3	4	6
Avr. F-Score	0.7082	**0.7092**	0.7088	0.7082	0.7073

TABLE X
EVALUATIONS ON DIFFERENT THRESHOLDS θ

θ	0.5	0.6	0.7	0.8
Avr. F-Score	0.7074	0.7088	**0.7092**	0.7056

valid set of our SRF and other trackers, and we reach a similar conclusion as in DAVIS that our SRF shows comparable results considering both accuracy and time. Compared to SiamR-CNN [25], SRF scarifices some accuracy to pursue higher speed, and SRF surpasses SiamMask [11] for both seen and unseen categories. Results on two datasets have shown that semi-supervised VOS tasks can be accomplished by taking full advantage of existing VOT trackers.

C. Ablation Study

In this subsection, we conduct ablation studies of our SRF using the VOT2019-LT dataset.

1) Effectiveness of Different Components: We try different combinations of the three parts in our framework, and the results are shown in Table VIII. Since we use the simplified SiamR-CNN [25] in SRF, we also compare the original SiamR-CNN model and its combo with our refine module. On VOT2019-LT, the simplified version produces comparable results (0.656) to the original one (0.663), but with a much simpler structure and faster speed. SiamR-CNN shows its strength in the recall. TrDiMP [37] also achieves similar performance on F-score (0.653) but with a relatively high score on precision. By combining the SiamR-CNN and TrDiMP, we take advantage of their respective strengths, and such combination raise both precision and recall, resulting in an F-score of 0.692. Refine module is further cascaded, boosting the F-score by 1.5 percentage points to 0.707, which is a remarkable result.

2) Evaluation of Successive Frame Number K: In our switch mechanism, K is the number of successive well-tracked frames from the global re-detector, after which the local tracker would be awakened again. Table IX shows that the performance is maximized when $K = 2$. On short-term tracking scenarios, discriminative trackers like SuperDiMP [3] and TrDiMP [37] perform better than global methods such as SiamR-CNN [25]. When the target is found with high confidence, handing over the task from the re-detector to the local tracker achieves better performance, and this might be the reason why larger K leads to a drop in F-score. On the other hand, two successive frames are more secure than one frame to guarantee that the target is actually detected.

3) Evaluation of Threshold θ: θ is the confidence threshold above which a frame with its predicted bounding box is thought to be reliable and is counted into the previous K. Smaller θ might lead to inaccurate predictions. At the same time, an overly high threshold would delay the transfer from the re-detector to the tracker, not fully releasing the ability of the local tracker. Table X shows the evaluation results on four different θs. Among them, $\theta = 0.7$ is the optimal choice.

TABLE XI
RESULTS OF SRF ON VOT2019-LT UNDER DIFFERENT SPEED DESIGNATION

Speed Control Parameter	0	0.25	0.5	0.75	1
FPS	14.6	15.4	16.8	18.3	21.3
F-score	0.708	0.707	0.706	0.703	0.697

TABLE XII
TIME COMPARISON

Tracker	LTMU [12]	Global Track [45]	Siam R-CNN [25]	Keep-Track [17]	Xuan *et al.*'s [26]	SRF Ours
FPS	13	6	4.7	12.7	3.8	**14.6 (21.3)**
Device	2080Ti	TitanX	V100	2080Ti	2080Ti	TitanXp

D. Speed Analysis

We did experiments on five SCP values with a gap of 0.25, and the results are shown in Table XI. When the parameter is tuned to 1, the fastest version reaches a speed of 21+ FPS, with performance that is still not inferior to existing trackers.

As shown in (11), when the SCP moves towards 1, less frames with poor confidence score from the short-term tracker would be sent into the re-detector for better estimation. In other words, more frames would use the output from the short-term tracker instead of that from the re-detector as the input of the refiner, which leads to inferior results. As a result, we recommend setting the initial value of SCP to 0 for utilizing the full strength of our framework. SCP shows the importance of the global re-detector.

In Table XII, we compare the tracking speed of our SRF along with other representative long-term tracking frameworks in FPS. SRF is the fastest in all the methods despite the fact that we did not use the most advanced GPU devices. To the best of our knowledge, SRF is the first long-term tracker whose speed is continuously adjustable.

E. Reflections on Experimental Results

In our framework, the backbone networks of all the three components use the ResNet [73] architecture but with different number of layers. Reusing the backbone networks in the three parts may contribute to faster speed and less memory usage. We use multiple trackers in SRF. In [52], an agent sharing network fuses results from multiple cameras tracking one target, which is a setting similar to ours. The size of each tracker's search area is usually regarded as a hyper-parameter. The work [74] has shown that reinforcement learning has the capability of optimizing the search area of each frame in the VOS tasks. Such idea can be borrowed into our VOT tasks. Separating the distractors from the target using visual or motion saliency based approaches [75]–[77] can also be tested in the refine module.

V. Conclusion

This paper introduces SRF, a long-term tracking and segmentation framework with a local tracker, a global re-detector, and a refine module. Simplicity is one of the characteristics of our framework. Following the experimental results that adding new modules does not necessarily lead to better performance, we take full advantage of current advanced trackers from three complementary perspectives. We show that instead of designing new networks, combining existing works in a straightforward way can also reach remarkable performances on seven tracking benchmarks. Moreover, we show that semi-supervised VOS tasks can be accomplished by using existing VOT trackers. Thanks to the refine module, SRF can estimate pixel-wise object locations with its precise contour while taking only bounding boxes as the initial input. SRF is shown to achieve a good balance between time and accuracy on two representative segmentation datasets. The continuous adjustable speed control parameter boosts the entire tracking process to 20+FPS with minor performance loss.

References

[1] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

[2] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6182–6191.

[3] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7183–7192.

[4] Y. Cui, C. Jiang, L. Wang, and G. Wu, "Fully convolutional online tracking," *arXiv:2004.07109*, 2020. [Online]. Available: http://arxiv.org/abs/2004.07109

[5] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Know your surroundings: Exploiting scene information for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 205–221.

[6] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8971–8980.

[7] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[8] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 07, Feb. 2020, pp. 12 549–12 556.

[9] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 101–117.

[10] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable siamese attention networks for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6728–6737.

[11] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.

[12] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang, "High-performance long-term tracking with meta-updater," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6298–6307.

[13] B. Yan, H. Zhao, D. Wang, H. Lu, and X. Yang, "'skimming-perusal'tracking: A framework for real-time and robust long-term tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2385–2393.

[14] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. ˇCehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey *et al.*, "The sixth visual object tracking vot2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, Sep. 2018, pp. 1–52.

[15] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2019, pp. 1–36.

[16] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K. Kämäräinen, M. Danelljan, L. Č. Zajc, A. Lukežič, O. Drbohlav *et al.*, "The eighth visual object tracking vot2020 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, Aug. 2020, pp. 547–601.

[17] C. Mayer, M. Danelljan, D. P. Paudel, and L. Van Gool, "Learning target candidate association to keep track of what not to track," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13 444–13 454.

[18] Y. Zhang, B. Ma, J. Wu, L. Huang, and J. Shen, "Capturing relevant context for visual tracking," *IEEE Trans. Multimedia*, vol. 23, pp. 4232–4244, Nov. 2020.

[19] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, no. 1-2, pp. 239–263, May 2002.

[20] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5374–5383.

[21] H. Fan, H. Bai, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, M. Huang, J. Liu, Y. Xu *et al.*, "Lasot: A high-quality large-scale single object tracking benchmark," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 439–461, Sep. 2020.

[22] A. Moudgil and V. Gandhi, "Long-term visual object tracking benchmark," in *Proc. Asian Conf. Comput. Vis.*, Dec. 2018, pp. 629–645.

[23] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. W. Smeulders, P. H. Torr, and E. Gavves, "Long-term tracking in the wild: A benchmark," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 670–685.

[24] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 445–461.

[25] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, "Siam r-cnn: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6578–6588.

[26] S. Xuan, S. Li, Z. Zhao, L. Kou, Z. Zhou, and G.-S. Xia, "Siamese networks with distractor-reduction method for long-term visual object tracking," *Pattern Recognition*, vol. 112, p. 107698, Apr. 2021.

[27] S. Choi, J. Lee, Y. Lee, and A. Hauptmann, "Robust long-term object tracking via improved discriminative model prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 602–617.

[28] H. Wu, X. Yang, Y. Yang, and G. Liu, "Flow guided short-term trackers with cascade detection for long-term tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2019, pp. 1–9.

[29] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4591–4600.

[30] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 771–787.

[31] X. Li, L. Huang, and Z. Wei, "A twofold convolutional regression tracking network with temporal and spatial mechanism," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1537–1551, Mar. 2022.

[32] Z. Zhou, X. Li, T. Zhang, H. Wang, and Z. He, "Object tracking via spatial-temporal memory network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2976–2989, May 2022.

[33] B. Yan, X. Zhang, D. Wang, H. Lu, and X. Yang, "Alpha-refine: Boosting tracking performance by precise bounding box estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5289–5298.

[34] B. Liao, C. Wang, Y. Wang, Y. Wang, and J. Yin, "Pg-net: Pixel to global matching network for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 429–444.

[35] G. Wang, C. Luo, X. Sun, Z. Xiong, and W. Zeng, "Tracking by instance detection: A meta-learning approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6288–6297.

[36] M. Jiang, Y. Zhao, and J. Kong, "Mutual learning and feature fusion siamese networks for visual object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3154–3167, Aug. 2020.

[37] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1571–1580.

[38] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8126–8135.

[39] H. Zhang, L. Cheng, T. Zhang, Y. Wang, W. Zhang, and J. Zhang, "Target-distractor aware deep tracking with discriminative enhancement learning loss," *IEEE Trans. Circuits Syst. Video Technol.*, 2022, early access.

[40] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Dec. 2011.

[41] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas, and M. Kristan, "Fucolot–a fully-correlational long-term tracker," in *Proc. Asian Conf. Comput. Vis.*, Aug. 2018, pp. 595–611.

[42] X. Wang, Z. Chen, J. Tang, B. Luo, Y. Wang, Y. Tian, and F. Wu, "Dynamic attention guided multi-trajectory analysis for single object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4895–4908, Dec. 2021.

[43] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kämäräinen, H. J. Chang, M. Danelljan, L. Cehovin, A. Lukežič *et al.*, "The ninth visual object tracking vot2021 challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2711–2738.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015.

[45] L. Huang, X. Zhao, and K. Huang, "Globaltrack: A simple and strong baseline for long-term tracking," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, Feb. 2020, pp. 11 037–11 044.

[46] F. Pernici and A. Del Bimbo, "Object tracking by oversampling local features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2538–2551, Dec. 2013.

[47] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 749–758.

[48] X. Li, L. Zhao, W. Ji, Y. Wu, F. Wu, M.-H. Yang, D. Tao, and I. Reid, "Multi-task structure-aware context modeling for robust keypoint-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 915–927, Apr. 2018.

[49] N. Wang, W. Zhou, and H. Li, "Reliable re-detection for long-term tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 730–743, Mar. 2018.

[50] F. Tang and Q. Ling, "Contour-aware long-term tracking with reliable re-detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4739–4754, Dec. 2019.

[51] B. Ramesh, S. Zhang, H. Yang, A. Ussa, M. Ong, G. Orchard, and C. Xiang, "e-tld: Event-based framework for dynamic object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3996–4006, Oct. 2020.

[52] P. Zhu, J. Zheng, D. Du, L. Wen, Y. Sun, and Q. Hu, "Multi-drone-based single object tracking with agent sharing network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 4058–4070, Oct. 2020.

[53] S.-Y. Chien, W.-K. Chan, Y.-H. Tseng, and H.-Y. Chen, "Video object segmentation and tracking framework with improved threshold decision and diffusion distance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 6, pp. 921–934, Jun. 2013.

[54] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.

[55] A. Lukezic, J. Matas, and M. Kristan, "D3s-a discriminative single shot segmentation tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7133–7142.

[56] A. Robinson, F. J. Lawin, M. Danelljan, F. S. Khan, and M. Felsberg, "Learning fast and robust target models for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7406–7415.

[57] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "Sstvos: Sparse spatiotemporal transformers for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5912–5921.

[58] Y. Zhang, Z. Wu, H. Peng, and S. Lin, "A transductive approach for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6949–6958.

[59] X. Chen, Z. Li, Y. Yuan, G. Yu, J. Shen, and D. Qi, "State-aware tracker for real-time video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9384–9393.

[60] G. Bhat, F. J. Lawin, M. Danelljan, A. Robinson, M. Felsberg, L. Van Gool, and R. Timofte, "Learning what to learn for video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 777–794.

[61] L. Wen, D. Du, P. Zhu, Q. Hu, Q. Wang, L. Bo, and S. Lyu, "Detection, tracking, and counting meets drones in crowds: A benchmark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7812–7821.

[62] P. Huang, J. Han, N. Liu, J. Ren, and D. Zhang, "Scribble-supervised video object segmentation," *IEEE/CAA J. of Automatica Sinica*, vol. 9, no. 2, pp. 339–353, Feb. 2021.

[63] D. Zhang, J. Han, L. Yang, and D. Xu, "Spftn: A joint learning framework for localizing and segmenting objects in weakly labeled videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 475–489, Feb. 2018.

[64] C. Chen, H. Wang, Y. Fang, and C. Peng, "A novel long-term iterative mining scheme for video salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, 2022, early access.

[65] J. Zhao, K. Dai, D. Wang, H. Lu, and X. Yang, "Online filtering training samples for robust visual tracking," in *Proc. of the 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1488–1496.

[66] 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 *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 740–755.

[67] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6154–6162.

[68] J. Luiten, P. Voigtlaender, and B. Leibe, "Premvos: Proposal-generation, refinement and merging for video object segmentation," in *Proc. Asian Conf. Comput. Vis.*, Dec. 2018, pp. 565–580.

[69] A. Lukeźič, L. Č. Zajc, T. Vojíř, J. Matas, and M. Kristan, "Performance evaluation methodology for long-term single-object tracking," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 6305–6318, Apr. 2020.

[70] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.

[71] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[72] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, "Youtube-vos: A large-scale video object segmentation benchmark," *arXiv:1809.03327*, 2018. [Online]. Available: http://arxiv.org/abs/1809.03327

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

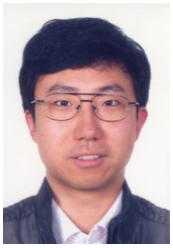
[74] J. Han, L. Yang, D. Zhang, X. Chang, and X. Liang, "Reinforcement cutting-agent learning for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 9080–9089.

[75] L. Jiang, Z. Wang, M. Xu, and Z. Wang, "Image saliency prediction in transformed domain: A deep complex neural network method," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, Jan. 2019, pp. 8521–8528.

[76] F. Zhou, R. Yao, G. Liao, B. Liu, and G. Qiu, "Visual saliency via embedding hierarchical knowledge in a deep neural network," *IEEE Trans. Image Process.*, vol. 29, pp. 8490–8505, Aug. 2020.

[77] R. R. A. Pramono, Y.-T. Chen, and W.-H. Fang, "Spatial-temporal action localization with hierarchical self-attention," *IEEE Trans. Multimedia*, vol. 24, pp. 625–639, Feb. 2021.

Xiang Xu received the B.Sc. degree in computer science and technology from Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 2020. He is currently pursuing the M.Sc. degree in computer science and technology from School of Artificial Intelligence, Nanjing University. His current research interests include object tracking and automatic annotation.



Jian Zhao received the B.Sc. degree from Nanjing University, Nanjing, China, in 2001, the M.Sc. degree from Hamburg University of Technology, Hamburg, Germany, in 2004, and the Ph.D. degree in electrical engineering from Swiss Federal Institute of Technology (ETH) Zürich, Zürich, Switzerland, in 2010. From 2010 to 2015, he was with the Institute for Infocomm Research, A*STAR, Singapore. He is currently an Associate Professor with the School of Electronic Science and Engineering, Nanjing University. His research interests include deep neural networks, mathematical optimization, and wireless communications. Dr. Zhao was honored with the Dengfeng Scholars Program of Nanjing University in 2015, the 2009 Chinese Government Award for Outstanding Self-Financed Students Abroad, and the IEEE Globecom 2008 Best Paper Award.

Jianmin Wu received the B.Sc. degree and M.Sc. degree from Shanghai Jiao Tong University, Shanghai, China. He is a Senior Engineer in State Grid Shanghai Maintenance Company, Shanghai. He has been engaged in the operation and maintenance of EHV and UHV power transmission and transformation equipment since 1996. He is proficient in high voltage electrical equipment related technology.

Furao Shen received the B.Sc. and M.Sc. degrees in mathematics from Nanjing University, Nanjing, China, in 1995 and 1998, respectively, and the Ph.D. degree from Tokyo Institute of Technology, Tokyo, Japan, in 2006. He is currently a Full Professor with Nanjing University. His current research interests include neural computing and robotic intelligence.