State-Aware Tracker for Real-Time Video Object Segmentation

Xi Chen[1]†, Zuoxin Li[2], Ye Yuan[2], Gang Yu[2], Jianxin Shen[1], Donglian Qi[1]*
[1] College of Electrical Engineering, Zhejiang University, [2]Megvii Inc.
{xichen_zju,J_X_Shen,qidl}@zju.edu.cn,
{lizuoxin,yuanye,yugang}@megvii.com

Abstract

In this work, we address the task of semi-supervised video object segmentation (VOS) and explore how to make efficient use of video property to tackle the challenge of semi-supervision. We propose a novel pipeline called State-Aware Tracker (SAT), which can produce accurate segmentation results with real-time speed. For higher efficiency, SAT takes advantage of the inter-frame consistency and deals with each target object as a tracklet. For more stable and robust performance over video sequences, SAT gets awareness for each state and makes self-adaptation via two feedback loops. One loop assists SAT in generating more stable tracklets. The other loop helps to construct a more robust and holistic target representation. SAT achieves a promising result of 72.3% $\mathcal{J}\&\mathcal{F}$ mean with 39 FPS on DAVIS2017-Val dataset, which shows a decent trade-off between efficiency and accuracy.

1. Introduction

Semi-supervised video object segmentation (VOS) requires to segment target objects over video sequences with only the initial mask given, which is a fundamental task for computer vision. In VOS task, the initial mask is provided as visual guidance. Nevertheless, throughout a video sequence, the target object can undergo large pose, scale, and appearance changes. Moreover, it can even meet abnormal states like occlusion, fast motion, and truncation. Therefore, it is a challenging task to make a robust representation over video sequences in a semi-supervised manner.

Luckily, video sequence brings additional context information for VOS task. First, the inter-frame consistency of video makes it possible to pass information efficiently between frames. Furthermore, in VOS tasks, information from preceding frames could be regarded as the temporal context, which can provide helpful cues for the following predictions. Hence, making efficient use of the additional information brought by video is of great importance for VOS tasks.

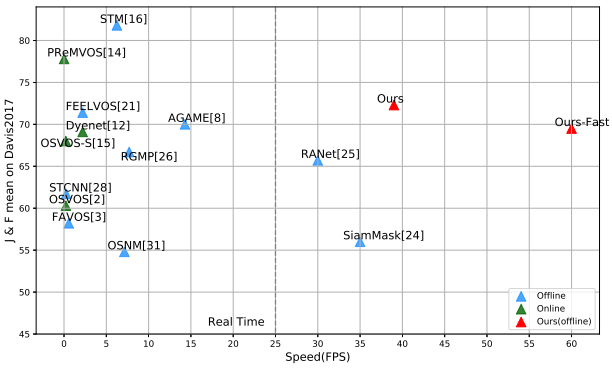

Figure 1. Accuracy versus speed on DAVIS2017-Val dataset. Some previous methods achieve high accuracy with slow running speed. Others sacrifice too much accuracy for the faster speed. Our method achieves a decent speed-accuracy trade-off.

However, previous works do not make good use of the characteristics of videos. [2, 15, 23, 26, 12] completely ignore the relation between frames and deal with each frame independently, which causes tremendous information waste. Other methods [22, 17, 31, 27, 31] use feature concatenation, correlation, or optical flow to propagate predicted mask or feature from the previous frame to the current frame, but they have apparent drawbacks. First, previous works usually propagate information on full images, while the target object usually occupies a small region. In this case, operations on full images can cause redundant computation. Furthermore, the target object can undergo different states throughout the video, but these methods apply fixed propagation strategies without adaptation, which makes them unstable over long sequences. Moreover, they only seek cues from the first or the previous frame for target modeling, which is not enough for a holistic representation. As a result, most existing methods can not tackle VOS with both satisfactory accuracy and fast speed. Therefore, a more efficient and robust pipeline for semi-supervised video ob-

*Corresponding Author
†This work was done during an internship at Megvii Inc.

ject segmentation is required.

In this paper, we reformulate VOS as a continuous process of state estimation and target modeling, in which segmentation is a specific aspect of state estimation. Specifically, we propose a simple and efficient pipeline called State-Aware Tracker (SAT). Taking advantage of the inter-frame consistency, SAT takes each target object as a tracklet, which not only makes the pipeline more efficient but also filters distractors to facilitate target modeling. In order to construct a more reliable information flow, we propose an estimation-feedback mechanism that enables our model to be aware of the current state and make self-adaptation for different states. For a more holistic target modeling, SAT uses the temporal context to construct a global representation dynamically to provide robust visual guidance throughout the video sequence. As demonstrated in Fig. 1, SAT achieves competitive accuracy and runs faster than all other approaches on DAVIS2017-Val dataset.

A simplified illustration of our pipeline is provided in Fig. 2. The inference procedure could be summarized as *Segmentation - Estimation - Feedback*. First, SAT crops a search region around the target object and takes each target as a tracklet. Joint Segmentation Network predicts masks for each tracklet. Second, State Estimator evaluates the segmentation result and produces a state score to represent the current state. Third, based on state estimation results, we design two feedback loops. Cropping Strategy Loop picks different methods adaptively to predict a bounding box for the target. Then, we crop the search region for the next frame according to the predicted box. This switching strategy makes the tracking process more stable over time. Meanwhile, Global Modeling Loop uses the state estimation results to update a global feature dynamically. In return, the global feature can assist Joint Segmentation Network in generating better segmentation results.

To verify the effectiveness of our method, we conduct extensive experiments and ablation studies on DAVIS2016, DAVIS2017 and YouTube-VOS datasets. Results show that SAT achieves strong performance with a decent speed-accuracy trade-off. Our main contributions can be summarized as follows: (1) We re-analyze the task of semi-supervised video object segmentation and develop State-Aware Tracker, which reaches both high accuracy and fast running speed on DAVIS benchmarks. (2) We propose a state estimation-feedback mechanism to make the VOS process more stable and robust over time. (3) We propose a new method of constructing global representation for the target object to provide more robust guidance.

2. Related Works

Video object segmentation task aims at segmenting target object in video frames given the initial mask of the first frame. In recent years, a wide variety of methods has been proposed to address this challenge. **Online learning based methods** : In order to distinguish the target object from background and distractors, online-learning based methods fine-tune the segmentation network on the first frame. OSVOS [2] fine-tunes a pretrained segmentation network on the first frame of test videos. OnAVOS [23] extends OSVOS by developing an online adaptation method. OSVOS-S [15] introduces instance information to enhance the performance of OSVOS. Lucid tracker [9] studies the data augmentation method for the first frame of test videos and brings significant improvement. Many other methods [25, 14, 32] take online learning as a boosting trick to reach better accuracy. Online learning has been proved to be an effective way to make VOS models more discriminative for the target object. However, it is too computational expensive to be used in practical applications. Generally, online models address the challenge of semi-supervised learning via updating model weight, which entails extensive iterations of optimization. Instead of updating model weight, our method updates a global representation via dynamic feature fusion, which tackles the challenge of target modeling more efficiently.

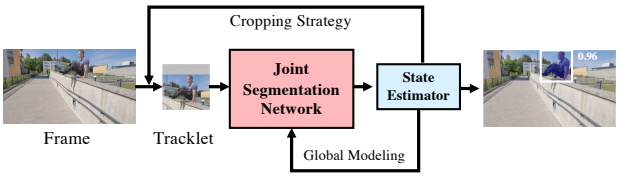

Figure 2. A simplified demonstration of our video object segmentation pipeline.

Offline learning based methods : Offline methods exploit the use of the initial frame and pass target information to the following frames via propagation or matching. MaskTrack [17] concatenates the predicted mask of the previous frame with the image of the current frame to provide spatial guidance. FEELVOS [22] develops pixel-wise correlation to pass location-sensitive embeddings over consecutive frames. RGMP [26] uses a siamese encoder to capture local similarities between the search image and the reference image. AGAME [8] proposes a probabilistic generative model to predict target and background feature distributions. These methods do not entail computational expensive online fine-tuning, but they still cannot reach fast speed due to inefficient information flow. Moreover, They usually suffer sub-optimal accuracy because they lack robust target representation. Our method is also offline trained and propagates visual cues from frame to frame. Different from previous, we take each object as a tracklet and apply self-adaptation, thus making the information flow more efficient and stable. Besides, we use the temporal context to update a global representation, which provides more robust guidance

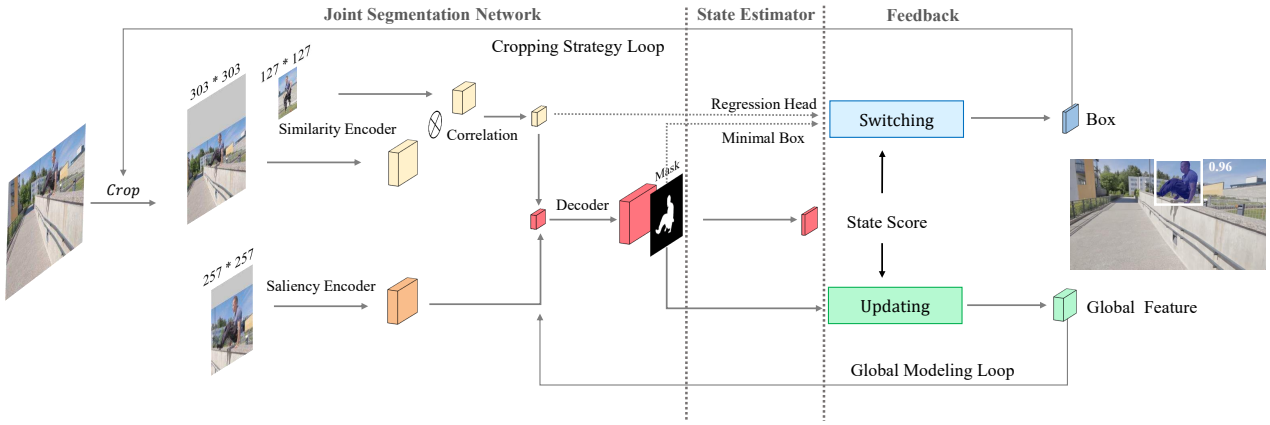

Figure 3. An overview of our video object segmentation pipeline. SAT can be divided into three parts by the dotted line in gray: Joint Segmentation Network, State Estimator, and Feedback. **Joint Segmentation Network** fuses the feature of the saliency encoder (in orange), the similarity encoder (in yellow), and the global feature (in green), and then decodes the fused feature to predict a mask. Afterward, **State Estimator** evaluates the prediction result and calculates a state score to represent the current state. Finally, based on the state estimation result, **Cropping Strategy Loop** switches the cropping strategy to keep a more stable tracklet. **Global Modeling Loop** constructs a global representation to enhance the feature of the segmentation network.

over video sequences.

Tracking based methods: FAVOS [3] develops a part-based tracking method to track local regions of the target object. SiamMask [24] narrows the gap between object tracking and object segmentation by adding a mask branch on SiamRPN [11], and it runs much faster than previous works. These tracking-based methods take tracking and segmentation as two separated parts. The segmentation result is not involved in the process of tracking, and it could be regarded as post-processing for the tracker. Different from previous works, we fuse object tracking and segmentation into a truly unified pipeline, in which there is no restrict boundary between tracking and segmentation. In our framework, these two tasks cooperate closely and enhance each other.

3. Method

3.1. Network Overview

In this work, we propose a novel pipeline called State-Aware Tracker (SAT), which gets high efficiency via dealing with each target as a tracklet. Besides, SAT gets awareness for each states and develop self-adaptation via two feedback loops.

As in Figure 3, we describe our inference procedure with three steps : **Segmentation - Estimation - Feedback.** First, *Joint Segmentation Network* fuses the feature of the similarity encoder, the saliency encoder, and the global feature to produce a mask prediction. Second, *State Estimator* evaluates the segmentation result and describes the current state with a state score and estimates whether it is a normal state or an abnormal state. Third, we construct two feedback loops to make self-adaptation for different states. In *Cropping Strategy Loop*, if it is a normal state, we use the predicted mask to generate a minimal bounding box. Otherwise, we use a regression head to predict the bounding-box and apply temporal smoothness. Then, based on the predicted box, we crop the search region for the next frame. In *Global Modeling Loop*, we use the state estimation results, the predicted mask and the current frame image patch to update a global feature, and use the global feature to enhance *Joint Segmentation Network* for better segmentation results. In the following section, we introduce each stage in detail.

3.2. Segmentation

As shown in Figure 3, the branch on bottom denotes the saliency encoder, and the two branches on top demonstrate the similarity encoder. For the input of the saliency encoder, we crop a relatively small region around the target to filter distractors, and we zoom it to a larger resolution to provide more details. In this way, the saliency encoder can extract a clean feature with rich details for the salient object of the input image patch. In this work, we use a shrinked ResNet-50 [6] for the saliency encoder.

The similarity encoder takes a larger search region of the current frame and a target region of the initial frame as input. It uses feature correlation to encode appearance similarities between the current image and the target object. This **correlated feature** provides appropriate supplementary for the saliency encoder to distinguish the target object from distractors. In this work, the implementation of the similarity encoder follows SiamFC++ [30] with Alexnet [10] backbone.

The saliency encoder extracts a class-agnostic feature for the target object, which is clean but lacks discrimination. Meanwhile, the correlated feature of the similarity encoder provides instance-level appearance similarity, which assists our network to distinguish the target object from distractors. In addition, the global feature updated by the Global Modeling Loop provides a holistic view for the target object, which is robust for visual variants over long sequences. In Joint Segmentation Network, we fuse these three features via element-wise addition to obtain a strong high-level feature with both discrimination and robustness.

After the feature fusion, we upsample the high-level feature by bilinear interpolation and concatenate it with low-level features of the saliency encoder successively. Consider that the input image of the saliency encoder is cropped around the target with high resolution, the low-level feature of the saliency encoder is clean and full of details, which assists the Joint Segmentation Network to decode a high-quality mask with fine contours.

3.3. Estimation

During the process of video segmentation, the target object can go through various states, such as well-presented, truncated, occluded, even can run out of the search region. In different states, we should take different actions to crop the search region for the next frame and apply different strategies to update the global representation.

State Estimator evaluates each local state with a state score and divides all states into two categories: normal state and abnormal state. We analyze that the state of the target object could be described by the mask predicting confidence and the mask concentration. As shown in Tab. 1, when the target is well-presented in the current image, the mask predicting confidence tends to be high, and the predicted mask is usually spatially concentrated. When the target gets truncated, the predicted mask tends to be separated into several parts, and it leads to low spatial concentration. When the target is occluded or runs out of the search region, the model usually predicts with low confidence.

	Confidence	*Concentration*	*State*
Well-Presented	High	High	Normal
Truncated	-	Low	Abnormal
Occluded	Low	-	Abnormal
Disappear	Low	-	Abnormal

Table 1. State estimation criterion. - denotes that the result does not influence the state estimation, which can be either high or low in this case.

Therefore, we propose a confidence score $\mathcal{S}_{cf}$ to denote the mask predicting confidence, and a concentration score $\mathcal{S}_{cc}$ to represent the geometric concentration for the predicted mask. We calculate the confidence score as Eq. 1, where $\mathcal{P}_{i,j}$ denotes mask prediction score at location (i, j), and $\mathcal{M}$ represents predicted binary mask. $\mathcal{M}_{i,j}$ equals 1 when the pixel at (i, j) is predicted as foreground, otherwise it equals 0.

$$\mathcal{S}_{cf} = \frac{\sum_{i,j} \mathcal{P}_{i,j} \cdot \mathcal{M}_{i,j}}{\sum_{i,j} \mathcal{M}_{i,j}} \tag{1}$$

We define concentration score as the ratio of the max connected region area to the total area of the predicted binary mask. As in Eq. 2, $|\mathcal{R}_i^c|$ denotes the pixel number of the *i th* connected region of the predicted mask.

$$\mathcal{S}_{cc} = \frac{max(\{|\mathcal{R}_1^c|, |\mathcal{R}_2^c|, \cdots, |\mathcal{R}_n^c|\})}{\sum_1^n |\mathcal{R}_i^c|} \tag{2}$$

Finally, we calculate a state score $\mathcal{S}_{state}$ as Eq. 3. If $S_{state} > \mathcal{T}$, we estimate the current state as a normal state. Otherwise, we judge it as an abnormal state. In this work, we set $\mathcal{T} = 0.85$ according to the result of the grid search.

$$\mathcal{S}_{state} = \mathcal{S}_{cf} \times \mathcal{S}_{cc} \tag{3}$$

3.4. Feedback

Based on the estimation result, we construct two feedback loops. One loop switches the cropping strategy to make our tracker more stable over time. The other loop updates a global representation to enhance the process of segmentation.

Cropping Strategy Loop: For each frame, we generate a bounding box for the target object and crop the search region for the next frame according to the box. In order to maintain a stable and accurate tracklet, we design two box generation strategies and switch the strategy for different states. For normal states, we select the largest connected region of the binary mask and calculate its minimal bounding box to indicate the position of the target. We use the largest connected region in order to avoid the interference of small pieces of false-positive predictions. For abnormal states, we add a regression head after the similarity encoder to predict a bounding box, then apply a temporal smoothness on location, scale, and ratio. In this work, we construct our regression head following SiamFC++[30].

Considering that mask can provide a more accurate representation for object contours when the object is well-presented, mask-box can predict a more accurate location in normal states. Furthermore, the mask-box corresponds to a smaller search region, which makes it more robust for distractors. In contrast, regression-box is generated from a larger search region, so it can retrieve the object when it runs fast. When the object is truncated, the regression-box can provide complete predictions for the target object. In addition, with the help of the temporal smoothness, the

regression-box can still indicate a reasonable location if the object is occluded or even disappeared.

With the above analysis, during inference, we pick mask-box for normal states to produce more accurate locations, while we choose regression-box for abnormal states to get more robust predictions. Fig. 4 demonstrates some examples for strategy switching. If we use mask-box for all frames, our model will lose track of the target when some abnormal states occur, otherwise if we keep using regression-boxes, we would get less accurate location predictions when the target is well-presented, or there are distractors in the background. Therefore, switching between these two strategies enables our model to make self-adaptions in different states and make our tracking process more accurate and stable.

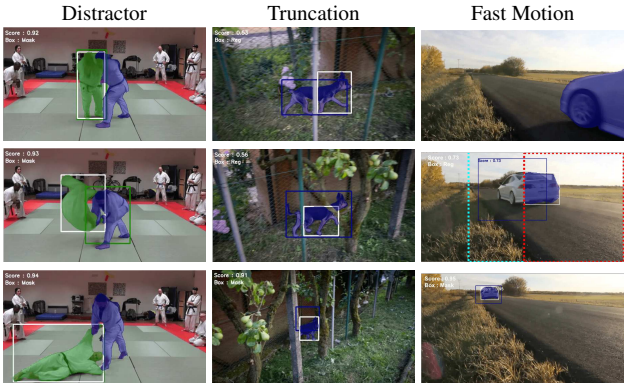

Figure 4. Switches between the mask-box (in white) and the regression-box (in color). The first column shows that the mask-box is more robust to distractors. When the two players are twisted together (second row), regression-box fails, and State Estimator chooses mask-box. The second column shows the regression-box provides a complete representation when the object is truncated or partially occluded. The third column shows that the regression-box can retrieve the target object in case of fast motion. The dotted line in cyan represents the search region of the similarity encoder; the one in red indicates the input region of the saliency encoder.

Global Modeling Loop : Global Modeling Loop updates a global feature for the target object dynamically, and uses this global feature to enhance the process of segmentation. As demonstrated in Fig. 5, after predicting the binary mask for frame T of target tracklet , we filter the background via element-wise multiplication. Then we feed the background-filtered image to a feature extractor (shrinked ResNet-50) to get a neat target feature. Consider that all background-filtered frames share the same instance-level content, in spite that the appearance of the target object could change violently through the video flow. We fuse the high-level features of each background-filtered frame step by step to updates a robust global representation. As Eq. 4, $\mathcal{G}$ denotes the global representation, and $\mathcal{F}$ denotes the high-level feature of the background-filtered image. μ denotes a hyper-parameter for step length that we set 0.5. Consider that if the target is occluded, disappeared, or poorly segmented, the extracted feature would be useless or even harmful for the global representation. Therefore, we score the high-level feature of each frame with the state score $\mathcal{S}_{state}$ produced by State Estimator, thus alleviates adverse effects caused by abnormal situations or low-quality masks.

$$\mathcal{G}_t = (1 - \mathcal{S}_{state} \cdot \mu) \cdot \mathcal{G}_{t-1} + \mathcal{S}_{state} \cdot \mu \cdot \mathcal{F}_t \qquad (4)$$

In this way, Global Modeling Loop updates a global feature that is robust for visual variants over time. In return, we use this global feature to enhance the high-level representation of Joint Segmentation Network. This feedback loop makes our target representation more holistic and robust for long video sequences.

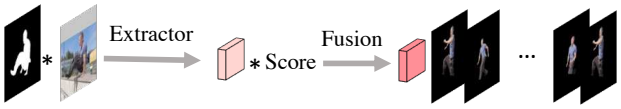

Figure 5. Updating process of Global Modeling Loop.

4. Experiments

4.1. Network Training

The whole training process consists of two stages. At the first stage, we train the similarity encoder and the regression head together on object tracking datasets [13, 4, 5, 7, 21]. The training strategy follows SiamFC++ [30]. Then, we train the whole pipeline with the weight of the similarity encoder and the regression head frozen. The backbone of the saliency encoder and the feature extractor in Global Modeling Loop are pretrained on ImageNet [4]. For training data, we adopt COCO [13], DAVIS2017 [20] training set (60 videos) and the YouTube-VOS [29] training set (3471 videos). We apply a cross-entropy loss on the predict binary mask of stride 4, and we also add auxiliary losses on the output feature of stride 8 (with weight 0.5) and stride 16 (with weight 0.3). We use SGD optimizer with momentum 0.9, set batch size as 16, and train our network on 8 GPUs with synchronized batch normalization. The training process takes about 8 hours, with 20 epochs. For each epoch, we select 160 000 images randomly. The first two epochs are a warm-up stage in which the learning rate increases linearly from 10^{-5} to 10^{-2}. In the last 18 epochs, we apply a cosine annealing learning rate.

For each iteration, we randomly choose one target image and one search image from the same video sequence. The saliency encoder takes the cropped search image as input, while Global Modeling Loop picks the cropped target image. We use the ground truth mask to filter the background

of the target image to train the extractor of Global Modeling Loop.

4.2. Ablation Studies

In Table 2, we perform extensive ablations on DAVIS2017 validation dataset. We upgrade our model step by step from the most naive baseline to the full-version SAT to verify the effectiveness of each principal component. Then, we also explore the upper-bound of our method.

Naive Seg Baseline : Our work starts from a naive segmentation baseline. We deal with each target as a tracklet, and we combine the saliency encoder and the decoder together to build a naive-segmentation network. For each video frame, we generate a min-max bounding box according to the predicted binary mask and crop a 257×257 search region for the next frame. This version performs weakly with only 48.1% $\mathcal{J}\&\mathcal{F}$ mean. When the target object is truncated, occluded, or run out of the search region, the min-max bounding box generated by the predicted mask cannot locate the target object, which causes target lost for successive frames.

Track-Seg Baseline: To tackle the problem of losing track. We combine a siamese tracker (SiamFC++[30]) and the naive-segmentation network together. We use the siamese tracker to predict the target location and use the naive-segmentation network to produce a binary mask. This version gains excellent improvement compared with the naive baseline. However, it is still not able to deal with large pose/scale variations, and the segmentation accuracy is heavily constrained by the tracking quality.

Correlated Feature: In order to obtain a more discriminative target representation, we introduce the correlated feature of similarity encoder to enhance the naive-segmentation network. The correlated feature contains appearance similarity, which brings 2.3% improvement.

Global Modeling Loop : For a more robust target representation over long sequences. We design Global modeling loop, which brings a significant improvement of 4.8%. The effectiveness of the mask filter and state score weight is shown in the second part of Table 2. Experiment results indicate that our idea of constructing a global representation is effective. Compared with only using the first frame or first frame + previous frame, the global representation brings 2.6% and 1.2% improvement respectively. We notice that the state score weight is also essential for updating the global representation, which improves the result by 1.2%. The effect of Global Modeling Loop is guaranteed by the mask filter, which brings a 5.6% improvement. We find that the version without mask filter and the version which concatenates mask filter with images both bring adverse effects. We analyze that foreground objects of different frames share the same high-level semantic representation despite pose or scale changes, while the background keeps changing through the whole video. Therefore, foreground features of different frames are complementary for each other, while background features are not additive. Hence, an explicit process of background filtering is necessary.

Version	CF	GM	CS	$\mathcal{J}\&\mathcal{F}$
Naive Seg				48.1
Track-Seg				61.6 (+13.5)
Track-Seg	✓			63.9 (+2.3)
Track-Seg	✓	✓		68.7 (+4.8)
Track-Seg (SAT)	✓	✓	✓	72.3 (+3.6)
first + previous frame	✓		✓	71.1 (-1.2)
first frame only	✓		✓	69.7 (-2.6)
no Score Weight	✓		✓	71.1 (-1.2)
no Mask Filter	✓		✓	66.7 (-5.6)
concat Mask	✓		✓	66.5 (-5.8)
Track-Seg	✓		✓	65.9 (-6.4)
Track-Seg		✓	✓	68.0 (-4.3)
Naive Seg	✓	✓		60.1 (-12.2)

Table 2. Ablation studies for each component on DAVIS2017-Val dataset. **CF** denotes Correlated Feature. **GM** denotes Global Modeling Loop. **CS** denotes Cropping Strategy Loop.

Cropping Strategy Loop: In order to maintain a more stable tracklet. We construct Cropping Strategy Loop, which switches the bounding box generation strategy according to the local state. This feedback loop brings a 3.6% improvement. More importantly, the switching mechanism weakens the dependency for either tracking results or segmentation results, which enables us to use small backbones for each branch.

We also analysis the switching mechanism by countering the usage rate of each strategy. On DAVIS2017-Val dataset, there are 30 sequences and 3923 frames in total. State Estimator judges 2876 (74%) frames as normal states 1047 (26%) as abnormal states. This statistic result agrees with our design intention that we use mask-box for the majority frames of normal states and regression-box for small numbers of abnormal situations.

Upper-Bound Analysis: As shown in Tab. 3, we explore the upper-bound of our pipeline by maximizing the effect of our two loops. For a clean global representation, we use the ground truth mask to filter the background of each frame, and this brings 1.7% improvement. For an accurate bounding box for search region cropping, we use the ground truth mask to generate minimal bounding box, which brings 1.8% improvement. In the ideal condition, the two loops make 5.2% improvement together. Therefore, constructing a robust global representation and maintaining a stable tracklet are two topics that worth further study.

	Mask Filter (GT)	Box (GT)	$\mathcal{J}\&\mathcal{F}$
SAT			72.3
SAT	✓		74.0 (+1.7)
SAT		✓	74.1 (+1.8)
SAT	✓	✓	77.5 (+5.2)

Table 3. Upper-Bound for our pipeline. Mask GT means using the ground truth mask to filter the background for global guidance. Box GT means using the ground truth bounding box to crop the search region for the next frame.

4.3. Comparison to state-of-the-art

We evaluate our method on DAVIS2017-Val [20], DAVIS2016-Val [18] and YouTube-VOS[29] datasets. Quantitative results demonstrate that our approach achieves promising performance for both accuracy and speed.

DAVIS2017: For the task of multi-object VOS, we predict a probability map for each target, then we concatenate them together, and apply a softmax aggregation to get the final result. We compare SAT with state-of-the-art methods. For the evaluation metrics, $\mathcal{J}\&\mathcal{F}$ evaluates the general quality of VOS result, $\mathcal{J}$ estimates the mask IOU, $\mathcal{F}$ describes the quality of contour. $\mathcal{J}_{\mathcal{D}}$ denotes the performance decay of $\mathcal{J}$ over time. FPS is measured for the time of every forward pass on a single RTX 2080Ti GPU.

As shown in Tab. 4, some newly proposed methods like FEELVOS [22], AGAME [8] aim to make the balance between speed and accuracy but SAT gets a more promising result for both. SiamMask [24] and RANet [25] also run at real-time speed, but their segmentation accuracy is obviously worse than ours. In general, SAT surpassed most of newly proposed models for both accuracy and efficiency.

SAT gets the best running speed and contour quality while achieves the highest $\mathcal{J}\&\mathcal{F}$ among newly proposed methods. Besides, SAT has the lowest performance decay $\mathcal{J}_{\mathcal{D}}$, which means our method is robust over time, and we would gain more advantages over others for long sequences. At the bottom row of Tab. 4, We also develop a faster version with ResNet-18 backbone, which runs at 60 FPS with slightly lower prediction accuracy.

YouTube-VOS: We mainly compare our method with some fast and offline learning methods on YouTube-VOS benchmark. Tab. 6 shows our method achieves competitive performance and surpasses [29, 26, 24] for both seen and unseen categories.

DAVIS2016: Single object segmentation is a relatively simpler task. As shown in Tab. 7, online fine-tuning often brings huge promotion on DAVIS2016 while costs enormous computation. Hence, we mainly compare our method with some newly proposed offline models. SAT performs better than FEELVOS [22], AGAME [8], RGMP [26] and SiamMask [24].

Computation Analysis: Running speed can be influenced by the environment and hardware condition. For a fair comparison, we also counter the multiply-accumulate operations of several fast VOS models. As shown in Tab. 5, our method costs obvious fewer Gflops than others. The computation of CNNs is highly related to input resolution and backbone size. Each component of SAT is specially designed for efficiency. The similarity encoder has a large input of 303×303, so we pick Alexnet as the backbone. The saliency encoder takes 257×257 image as input, and we use a shrinked ResNet-50 backbone, in which we set the channel expansion rate as 1. Global Modeling Loop only cares about the high-level feature, so we resize the filtered images to 129×129. In contrast, RANet [25] and AGAME [8] use ResNet-101 backbone with 480×864 input size, which makes them computational expensive. SiamMask [24] takes 255×255 images as input and uses a ResNet-50 backbone, and it replaces the stride-2 convolutions of the last two stages to stride-1, which helps to keep spatial information but brings more computation. Besides, SiamMask follows DeepMask[19] to apply a pixel-wise mask representation, which entails much computation.

Method	OL	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_{\mathcal{M}\uparrow}$	$\mathcal{J}_{\mathcal{D}\downarrow}$	$\mathcal{F}_{\mathcal{M}\uparrow}$	FPS
PReMVOS[14]	✓	77.8	73.9	16.2	81.7	0.01
OSVOS-s[15]	✓	68.0	64.7	15.1	71.3	0.22
OnAVOS[23]	✓	67.9	64.5	27.9	71.2	0.08
CINM[1]	✓	67.5	64.5	24.6	70.5	0.01
Dyenet[12]	✓	69.1	67.3	-	71.0	2.4
OSVOS[2]	✓	60.3	56.7	26.1	63.9	0.22
*STM[16]	×	**81.8**	**79.2**	-	**84.3**	6.25
FEELVOS [22]	×	71.5	69.1	17.5	74.0	2.2
AGAME[8]	×	70.0	67.2	**14.0**	72.7	14.3
RGMP[26]	×	66.7	64.8	18.9	68.6	7.7
RANet[25]	×	65.7	63.2	18.6	68.2	30
STCNN[28]	×	61.7	58.7	-	64.6	0.25
FAVOS[3]	×	58.2	54.6	14.4	61.8	0.56
SiamMask[24]	×	56.4	54.3	19.3	58.5	35
Ours	×	**72.3**	**68.6**	**13.6**	**76.0**	**39**
Ours-Fast	×	69.5	65.4	16.6	73.6	**60**

Table 4. Quantitative results on DAVIS2017 validation set. OL denotes online fine-tuning. FPS denotes frame per second. The best two results among offline methods are marked in red and blue respectively. *: STM requires more training data and longer training time than other works.

Method	Ours-f	Ours	SiamMask [24]	RANet [25]	AGAME [8]
Gflops	~ 12	~ 13	~ 16	> 65	> 65
FPS	60	39	35	30	14.3

Table 5. Computation analysis for some fast VOS models, Gflops counters multiply-accumulate operations. Ours-f denotes the fast version SAT with a Alexnet backbone.

Figure 6. Qualitative results of SAT on DAVIS Benchmark.

Method	OL	$\mathcal{G}$	$\mathcal{J}s$	$\mathcal{J}u$	$\mathcal{F}s$	$\mathcal{F}u$
PreMVOS[14]	✓	66.9	71.4	56.5	75.9	63.7
OSVOS[2]	✓	58.8	59.8	54.2	60.5	60.7
OnAVOS[23]	✓	55.2	60.1	46.1	62.7	51.4
*STM[16]	×	**79.4**	**79.7**	**84.2**	**72.8**	**80.9**
S2S[29]	×	57.6	66.7	48.2	-	-
RGMP [26]	×	53.8	59.5	45.2	-	-
SiamMask[24]	×	52.8	60.2	45.1	58.2	47.7
Ours	×	**63.6**	**67.1**	**55.3**	**70.2**	**61.7**

Table 6. Quantitative results on Youtube-VOS benckmark. OL denotes online fine-tuning. The subscript s denotes seen categories while u denotes unseen categories. The best two results among offline methods are marked in red and blue respectively. *: STM requires more training data and longer training time than other works.

Method	OL	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_{\mathcal{M}\uparrow}$	$\mathcal{F}_{\mathcal{M}\uparrow}$	FPS
RANet+[25]	✓	87.1	86.6	87.6	0.25
PReMVOS[14]	✓	86.8	84.9	88.6	0.01
OSVOS[2]	✓	80.2	79.8	80.6	0.22
*STM[16]	×	**89.3**	**88.7**	**89.9**	6.25
RGMP[26]	×	81.8	81.5	82.0	7.7
AGAME[8]	×	-	82.0	-	14.3
FEELVOS[22]	×	81.7	81.1	82.2	2.2
FAVOS[3]	×	80.8	82.4	79.5	0.56
SiamMask[24]	×	69.8	71.7	67.8	**35**
Ours	×	**83.1**	**82.6**	**83.6**	**39**

Table 7. Quantitative results on DAVIS2016 validation set. OL denotes online fine-tuning. FPS denotes frame per second. The best two results among offline methods are marked in red and blue respectively. *: STM requires more training data and longer training time than other works.

4.4. Qualitative result

Fig. 6 shows the qualitative result of our method on DAVIS benchmarks. SAT can produce robust and accurate segmentation results even in complicated scenes. The first three rows show that SAT is robust for distractors, motion blur and occlusion. The last row shows that SAT is robust for tremendous pose variant.

5. Conclusion

In this paper, we present State-Aware Tracker (SAT), which achieves promising performance with high efficiency for the task of semi-supervised video object segmentation. SAT takes each target object as a tracklet to perform VOS more efficiently. With an Estimation-Feedback mechanism, SAT can get awareness for the current state and make self-adaptation to reach stable and robust performance. Our methods achieves competitive performance on several VOS benchmarks with a decent speed-accuracy trade-off.

Acknowledgements: This paper is supported by the National key R&D plan of the Ministry of science and technology (Project Name: "Grid function expansion technology and equipment for community risk prevention", Project No. 2018YFC0809704)

References

[1] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5977–5986, 2018.

[2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

[3] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7415–7424, 2018.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5374–5383, 2019.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv preprint arXiv:1810.11981*, 2018.

[8] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2019.

[9] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for object tracking. In *The DAVIS Challenge on Video Object Segmentation*, 2017.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] 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*, pages 8971–8980, 2018.

[12] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 90–105, 2018.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[14] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580. Springer, 2018.

[15] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1515–1530, 2018.

[16] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9226–9235, 2019.

[17] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017.

[18] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

[19] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015.

[20] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[21] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5296–5305, 2017.

[22] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9481–9490, 2019.

[23] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In *The 2017 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*, volume 5, 2017.

[24] 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*, pages 1328–1338, 2019.

[25] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. *arXiv preprint arXiv:1908.06647*, 2019.

[26] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018.

[27] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video ob-

ject segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1140–1148, 2018.

[28] 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*, pages 1379–1388, 2019.

[29] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, 2018.

[30] Yinda Xu, Zeyu Wang, Zuoxin Li, Yuan Ye, and Gang Yu. SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines. *arXiv e-prints*, page arXiv:1911.06188, 2019.

[31] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018.

[32] Qiang Zhou, Zilong Huang, Xinggang Wang, Yongchao Gong, Han Shen, Lichao Huang, Chang Huang, and Wenyu Liu. Proposal, tracking and segmentation (pts): A cascaded network for video object segmentation. *arXiv preprint arXiv:1907.01203*, 2019.