Learning Spatio-Appearance Memory Network for High-Performance Visual Tracking

Fei Xie[1], Wankou Yang[1], Kaihua Zhang[2], Bo Liu[3], Guangting Wang[4], Wangmeng Zuo[5]
[1]School of Automation, Southeast University, China
[2]Nanjing University of Information Science and Technology, China
[3]JD Finance America Corporation, Mountain View, CA, USA
[4]University of Science and Technology of China
[5]School of Computer Science and Technology, Harbin Institute of Technology
jaffe03@seu.edu.cn, wkyang@seu.edu.cn, zhkhua@gmail.com, kfliubo@gmail.com
flylight@ustc.edu.cn, cswmzuo@gmail.com

Abstract

Segmentation-based tracking is currently a promising tracking paradigm due to the robustness towards non-grid deformations, comparing to the traditional box-based tracking methods. However, existing segmentation-based trackers are insufficient in modeling and exploiting dense pixel-wise correspondence across frames. To overcome these limitations, this paper presents a novel segmentation-based tracking architecture equipped with spatio-appearance memory networks. The appearance memory network utilizes spatio-temporal non-local similarity to propagate segmentation mask to the current frame, which can effectively capture long-range appearance variations and we further treat discriminative correlation filter as spatial memory bank to store the mapping between feature map and spatial map. Moreover, mutual promotion on dual memory networks greatly boost the overall tracking performance. We further propose a dynamic memory machine (DMM) which employs the Earth Mover's Distance (EMD) to reweight memory samples. Without bells and whistles, our simple-yet-effective tracking architecture sets a new state-of-the-art on six tracking benchmarks. Besides, our approach achieves comparable results on two video object segmentation benchmarks. Code and model are released at `https://github.com/phiphiphi31/DMB`.

1. Introduction

Visual object tracking (VOT) is a fundamental task in computer vision. In general, VOT aims at localizing the target in subsequent frames based the given bounding box in the first frame. So far, VOT still remains a challenging topic due to numerous factors such as deformation, occlusion, and background clutter [47, 18, 9]. Two dominant methodologies of deep VOTs, Siamese correlation networks [22, 21, 50, 4] and discriminative correlation filters (DCFs) [7, 1], mainly adopt a bounding box-level target representation, making them limited in exploiting the fine-scale representation of the target that is essential to achieve a high tracking accuracy. To address these issues, pixel-wise target estimation is needed, but it requires segmentation datasets [49] for training which is far less than the tracking datasets, such as TrackingNet [28], Lasot [9] and GOT-10K [15], due to the extremely laborious annotations.

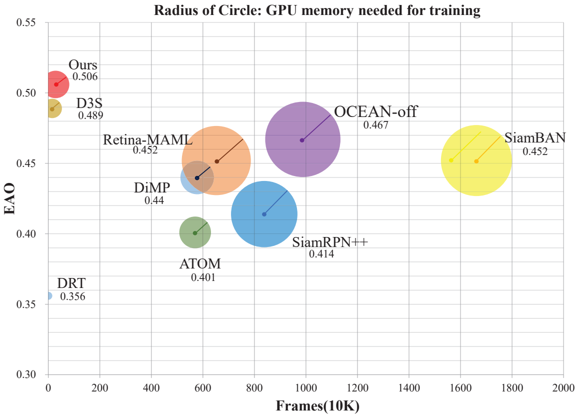

Figure 1. Comparison of tracking performance and offline training cost with state-of-the-art trackers on VOT2018 [18]. We visualize the Expected Average Overlap (EAO) with respect to the amount of training frames. The radius of circle denotes the GPU memory needed for training (16GB is needed for our tracker). DRT [39] is a fully online tracker that achieves the best efficiency but a much lower EAO than ours.

Several attempts have been made to develop segmentation-based trackers. SiamMask [44] adds a segmentation branch to Siamese architecture for allowing joint learning of bounding box regression and object segmentation in training. Later, D3S [27] introduces a segmentation branch following VideoMatch [13] and further combines online DCF [2] to fuse target classification and pixel-wise segmentation during inference. While the DCF can be updated to cope with appearance variations across frames, little study has been given to consider temporal information in the segmentation branch.

To overcome the limitations, this paper exploits spatio-appearance memory networks to capture long-range spatio-temporal information. We present the appearance memory network (AMN) for adapting the segmentation branch to temporal appearance variation while avoiding model drifting. In particular, we store keys and values of continuous frames in the AMN, and design a memory reader to compute the spatio-temporal attention to previous frames for each pixel in the query image (i.e., the current frame). Thus, albeit the network parameters of the memory networks are fixed, we can dynamically update the memory samples to achieve better tradeoff between model generalization and flexibility. We further treat DCF as spatial memory network (SMN) to model the mapping between feature map and spatial map. Moreover, the SMN can help to filter out the noisy samples in AMN while AMN provides SMN with more accurate target geometrical center. This mutual promotion on dual memory banks greatly boost the tracking performance.

In practice, the target object in the query frame usually appears in the local neighborhood of memory frames. So the importance of each sample stored in memory banks varies due to target deformation and background clutters. To solve the problems, we further propose a dynamic memory machine (DMM) to reweight the memory samples in both spatial and temporal domain. DMM employs the Earth Mover's Distance (EMD) to improve the memory reader module. It generates weighting values by obtaining the optimal matching flows between search and template patches. Moreover, DMM applys a background suppression mechanism in EMD computation which not only minimizes the impact of background clutters but still treats background as important context information.

Extensive experiments show that our tracker sets a new state-of-the-art on the popular tracking benchmarks including VOT2016, VOT2018, VOT2019, VOT2020, GOT-10K, and TrackingNet. Moreover, for the video object segmentation (VOS) task, our tracker also achieves comparable results on the DAVIS16 and DAVIS17 benchmarks. In comparison to template-based trackers, our approach can reduce the training data by more than an order of magnitude with improved tracking performance (See Fig. 1).

The main contributions of this work are three-fold:

- We propose a novel segmentation-based tracking architecture which uses appearance memory bank to capture and exploit the temporal appearance changes to enhance the segmentation branch.
- We firstly employ a novel background suppression mechanism into tracking problem and propose a dynamic memory machine applied to deep trackers with memory networks.
- Extensive experiments show that our approach achieves state-of-the-art results on six challenging tracking benchmarks and competitive results on DAVIS16 and DAVIS17 for VOS.

2. Related Work

2.1. Segmentation based VOT

Video object segmentation (VOS) [40, 5, 3] methods usually are slow in speed and are not effective on handling the challenging factors in tracking scenarios, e.g., distractors and fast motion. It is mainly because the VOS task considers segmentation of large objects with limited appearance changes in short videos. SiamMask [44] attempts to unify tracking and segmentation by adding a class-agnostic segmentation branch to detection-based tracker. SiamR-CNN [41] uses an well-trained segmentation model to estimate the mask in the box which considers the predicted bounding box as hard spatial constraints. Similarly, many VOT methods such as OceanPlus [17] and SiamMargin [17] add extra segmentation head after predicting bounding box to improve tracking accuracy. D3S [27] uses the DCF as the classification branch and a geometrically invariant template-based model for object segmentation. Comparing to the traditional bounding box based tracking methods [22, 21, 7, 1], our method adopts the segmentation tracking architecture, and the spatio-appearance memory networks are introduced to utilize temporal information.

2.2. Memory Network for Video Analysis

Memory network is a kind of neural network [10, 45] that have external memory where information can be stored and read by purpose. Recently, memory network has exhibited its merits in temporal modeling for video tasks. In visual tracking, MemTrack [52] uses a dynamic memory network to adapt the template to appearance variations. STM [30] applies memory networks to semi-supervised VOS and achieves appealing performance. However, most of those VOS methods typically consider large targets with low background distractor presence. In object tracking benchmarks, the scenarios are more challenging. In this work, we present spatio-appearance memory networks for both VOS and visual tracking tasks. Mutual promotions enhance the overall performance on tracking scenarios.

2.3. Earth Mover's Distance

EMD is suitable to compare structural representations without explicit alignment information. Zhao *et al*. [55]

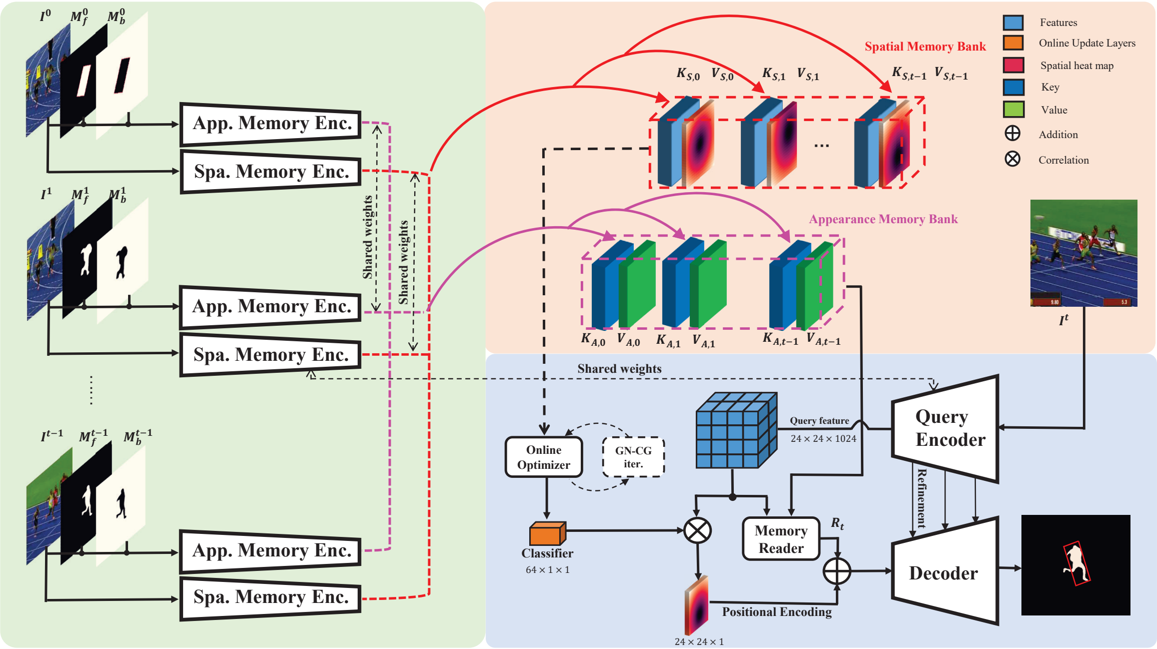

Figure 2. Overview of our segmentation-based tracking architecture with spatio-appearance memory networks. The model consists of two memory networks. One is SMN which is trained by online optimization. The other is AMN which makes dense non-local matching to capture stable appearance information. The fused read-out from the spatio-appearance memory networks are fed to decoder to predict the mask. Finally, the bounding box of the tracked target is estimated from the mask.

propose to calculate the differential EMD to solve the visual tracking problem. Li [23] uses a tensor-SIFT based EMD to solve the contour tracking problem. Zhang employs the Earth Mover's Distance (EMD) as a metric for few-shot image classification task. In this work, we firstly employ differential EMD to compute the temporal weights of each sample in memory banks in both visual tracking and VOS.

3. Proposed Method

In the following, we first introduce the overall pipeline of our approach in Section 3.1, and then explain the appearance memory network (Section 3.2), spatial memory network (Section 3.3) in details. The benefits of our spatio-appearance memory network design are explained in Section 3.4, and finally, we provide the design of the dynamic memory machine in Section 3.5.

3.1. Overall Pipeline

Fig. 2 illustrates the pipeline of our approach. Each frame I_t is embedded into two triplets $(Q_t, K_{A,t}, V_{A,t})$ and $(Q_t, K_{S,t}, V_{S,t})$. As in [46], Q, K, and V refer to Query, Key, and Value, respectively. For the tracking and segmentation of current frame I_t, an appearance memory encoder $\mathbf{Enc}_M^A$ is used to compute the appearance memory key and value representation pairs $\{(K_{A,0}, V_{A,0}), ..., (K_{A,t-1}, V_{A,t-1})\}$ for the previous frames $\{I_0, ..., I_{t-1}\}$. Meanwhile, a spatial memory encoder $\mathbf{Enc}_M^S$ is introduced to extract the spatial memory keys $\{K_{S,0}, ..., K_{S,t-1}\}$. Following conventional tracking setting, the values $\{V_{S,0}, ..., V_{S,t-1}\}$ of the spatial memory are computed based on the annotation of the first frame and the predicted target bounding boxes of previous frames. Moreover, a query encoder $\mathbf{Enc}_Q$ is designed to obtain the query Q_t and the query value $V_{Q,t}$ for the current frame I_t. Furthermore, the memory reader module is adopted to generate the value $V_{A,t}$ for the current frame. As for spatial memory, we take DCF as a memory module, and use it to generate target location map. Subsequently, $V_{A,t}$, $V_{Q,t}$, the target location map, and the query encoder features are fused into a decoder to predict the segmentation mask of I_t. Finally, the target bounding box can be estimated from the segmentation mask. To adapt target appearance variations over time during tracking, the memory keys and values are stored and updated online.

3.2. Appearance Memory Network

Fig. 3 shows the architecture of our appearance memory network that includes memory bank and a reader. Analogous to conventional memory network [45, 31], our memory network consists of memory encoder $\mathbf{Enc}_M^A$, query encoder $\mathbf{Enc}_Q$, and memory reader. In particular, for each of the previous frames, the memory encoder takes the image I and the foreground as well as the background segmentation masks $\{M_f, M_b\}$ as the input to produce the key and the value. And the current frame I_t is fed into query encoder

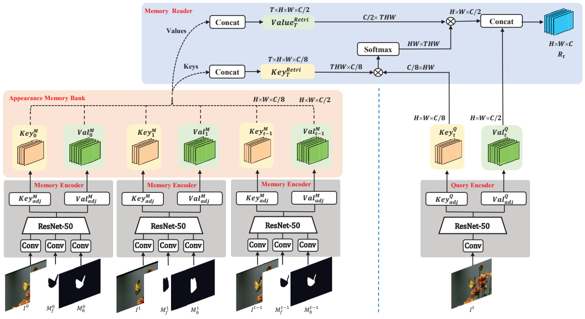

Figure 3. Overview of our appearance memory network. Each continuing frame and its foreground-background mask generates corresponding key and value through appearance memory encoder. Query frame I_t will be encoded into query Q_t and value $V_{A,t}$ embedding. A dense non-local matching operation will be performed between query Q_t and stored memory keys $\{K_{A,0}, ..., K_{A,t-1}\}$. The retrieved value $V_{Q,t}$ from read operation will be concatenated with query value $V_{A,t}$ as the read-out value R_t. Then, the read-out value R_t will be fed into decoder for final mask prediction.

to obtain query Q_t and query value $V_{Q,t}$. Then, query Q_t is passed into the memory reader to obtain the retrieved value $V_{A,t}$ from AMB. Finally, $V_{A,t}$ and $V_{Q,t}$ are concatenated to form the read-out value R_t. Next, we introduce the memory encoder, query encoder and memory reader in detail.

Memory Encoder. The input of memory encoder involves three components, i.e., an RGB frame, the foreground and background segmentation masks with probability between 0 and 1. Each component first goes through three convolution layers individually and then be summed and fed into the backbone. Here we take ResNet-50 [12] as the backbone for both the memory encoder and the query encoder, and use the Conv4_e layer as the common feature map f_M for computing the key and value. Then, the key and value can be obtained by respectively deploying their own convolution layer on the common feature map f_M,

$$K_A = \mathbf{Key}_A\left(f_M\right), \quad V_A = \mathbf{Val}_A\left(f_M\right). \tag{1}$$

During tracking, keys and values from all previous frames are stacked along the temporal order and are stored in the appearance memory bank.

Query Encoder. The query encoder $\mathbf{Enc}_Q$ takes the current frame I_t as the input. Analogous to memory encoder, we use the Conv4_e layer of ResNet-50 as the common feature map f_Q. To generate the query Q_t, a convolution layer with linear activation is applied to reduce the number of channels to the 1/8 of f_Q. The channel number of query value $V_{Q,t}$ is a half of f_Q,

$$Q_t = \mathbf{Que}_A(f_Q), V_{Q,t} = \mathbf{Val}_Q(f_Q). \tag{2}$$

Memory Reader. In the memory reader module, the keys and values $\{(K_{A,0}, V_{A,0}), ..., (K_{A,t-1}, V_{A,t-1})\}$ of all previous frames, and the query and query value $(Q_t, V_{Q,t})$ of the current frame are used to produce the read-out value R_t. In particular, the similarities between query Q_t and keys $\{K_{A,0}, ..., K_{A,t-1})\}$ are utilized to measure the spatial and temporal non-local correspondence, which is then used to generate the retrieved value $V_{A,t}$ for capturing temporal appearance changes. Then, the retrieved value $V_{A,t}$ is computed based on the non-local attention mechanism formulated as follows,

$$V_{A,t}^i = \sum_j \sum_{k=1}^{t-1} A_t^{i,j,k} V_{A,k}^j, \tag{3}$$

$$A_t^{i,j,k} = \frac{\exp\left\langle Q_t^i, K_{A,k}^j\right\rangle}{\sum_p \sum_{k=1}^{t-1} \exp\left\langle Q_t^i, K_{t-1}^p\right\rangle}, \tag{4}$$

where i, j, and p denote a spatial position of feature map, k denotes the index of a frame, and $\langle\cdot,\cdot\rangle$ denotes the dot product between two vectors. Furthermore, for enhancing the retrieved value $V_{A,t}$, we concatenate it with the query value $V_{Q,t}$ to obtain the read-out value,

$$R_t = concat\left[V_{A,t}, V_{Q,t}\right], \tag{5}$$

where $concat[\cdot,\cdot]$ denotes the concatenation operation. In contrast to MAST [20] where the RGB image or segmentation mask are adopted as the value, we predict the appearance value as embedding feature map. And we encode both the query and the query value, where later is further concatenated with the retrieved value to get the read-out value.

Decoder. The fused read-out from the spatio-appearance memory networks are fed to decoder to predict the mask. Please refer to [31] for the decoder design.

3.3. Spatial Memory Network

Inspired by [7], we treat DCF as the Spatial Memory Network (SMN) for target localization. The query encoder $\mathbf{Enc}_Q$ in AMN shares weights with the memory encoder and the query encoder for SMN. Let $x_k = \mathbf{Enc}_Q(I_k)$ be the feature map for a previous frame I_k, and y_k be the corresponding spatial label. The DCF model can then be formulated as,

$$f^* = \arg\min_f \sum_{k=0}^{t-1} \sum_p \|\langle x_k^p, f\rangle - y_k\|_2^2 + \lambda \|f\|_2^2. \tag{6}$$

The feature map of the current frame I_t is denoted by x_t. With the kernel tricks, we have,

$$R_{S,t}^i = \langle f^*, x_t^i\rangle = \sum_{k=0}^{t-1} \sum_j V_{S,k}^j A_{S,t}^{i,j,k}, \tag{7}$$

$$A_{S,t}^{i,j,k} = \left\langle x_t^i, x_k^j \right\rangle, \tag{8}$$

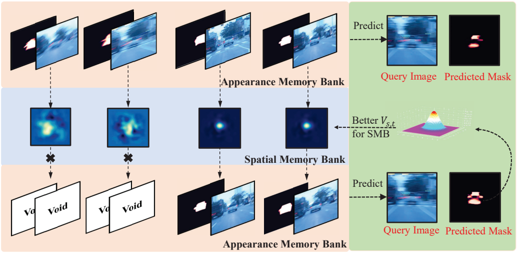

Figure 4. Visualization on benefits from filtered samples.

where i, j, and p denote a spatial position. We note that $\{x_k|k = 0, ..., t-1\}$ and x_t can be explained as the keys and query, while $V_{S,k}$ and $R_{S,t}$ are the value and read-out value in SMN. Thus, DCF can be explained as a special implementation of memory module to store the mapping between feature map and the read-out spatial map $R_{S,t}$. Moreover, the spatial map $R_{S,t}$ can serve as the spatial encoding of target localization, which is complementary to the read-out value in AMN. In our approach, we combine AMN and SMN to constitute the spatio-appearance memory banks for improving segmentation and tracking performance.

3.4. Benefit from Dual Memory Banks

In general, SMB is complementary to AMB and can collaborate to improve segmentation and tracking performance. We also present elaborate design to make the two banks cooperate well. The localization of SMB can be more robust because of the geometrical robustness of target center as shown in Fig. 3.

3.5. Dynamic Memory Machine

We apply the Earth Mover's Distance (EMD) to reweight memory frames by considering foreground and background similarities and organize all the features by a weighted averaging. In our design, the query feature of query frame $K_{Q,t}$ can be regarded as destination while the key features of memory frames $K_{M,i}, i \in \{1, 2, ..., t-1\}$ are as supplier. As shown in Fig. 5, after pooling layer, $K_{Q,t}$ and $K_{M,a}$ are both resized to 6 × 6. Grid cell positions of $K_{M,a}$ are supposed to be suppliers $\mathcal{S} = \{s_a|a = 1, 2, ..., 36\}$ and they required to transport goods to a set of destinations $\mathcal{D} = \{d_b|b = 1, 2, ..., 36\}$ which is composed of all positions of $K_{Q,t}$, where s_a denotes the supply units of supplier a and d_b represents the demand of b-th demander. The cost per unit transported from supplier a to demander b is denoted by c_{ab}, and the number of units transported is denoted by x_{ab}. The goal of the transportation problem is then to find a least-expensive flow of goods $\tilde{\mathcal{X}} = \{\tilde{x}_{ab}|a = 1, ...36, b = 1, ...36\}$ from the suppliers to the demanders: The supply units in s_a and demanding units in d_b denotes the importance of each grid cell in $K_{Q,T}$ and $K_{M,i}$. Actually, memory frames and query frame have larger background regions than the target object.

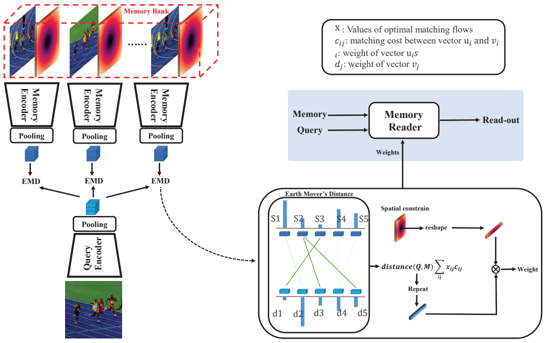

Figure 5. Overview of dynamic memory machine. The weights are computed among query frame and memory frames and further be applied to the memory reader.

Thus, large weights should be given to the foreground object region during the EMD calculation. However, memory frames whose background are similar to the query frame contains more valuable information. The ultimate goal of re-weight is not to completely eliminate the impact of background, but pay more attention to the foreground object. We observe that the co-occurrent regions in two images are more likely to be the foreground. On the case that two images have similar foreground and background, the distribution of weight values does not affect the distance severely. Therefore, we use dot product between a vector and the average feature in the other structure to generate a relevance score as the weight value:

$$s_a = \max\{\mathbf{u}_a^T \cdot \frac{\sum_{b=1}^{HW} \mathbf{v}_b}{HW}, 0\}, \tag{9}$$

where $\mathbf{u}_a$ and $\mathbf{v}_b$ denotes the vectors from $K_{Q,t}$ and $K_{M,i}$. H, W are the height, width. Then, we normalize all the weights to make equality of both sides:

$$s_a := s_a \frac{HW}{\sum_{b=1}^{HW} s_b}. \tag{10}$$

Similarly, the weights for each demander can be obtained in the same manner. Then, the weights of each memory samples will be added to the corresponding attention calculation.

4. Experiments

4.1. Implementation details

Training phase. For a better feature extraction ability, we firstly use image datasets instead of video sequences. ResNet-50 is initialized from the ImageNet pre-trained model. Similar to the training process in [32], we use image datasets annotated with object masks ([36], [25], [37], [24], [6]) to train our network. We apply image augmentations like random affine, flip and blur to the same image for generating a sequence of three images.

Trackers		SPM [43]	SiamMask-opt [44]	SaimRPN++ [21]	ATOM [7]	D3S [27]	Ours (SAMN)
	Acc.↑	0.62	0.67	0.64	0.61	0.66	0.684
VOT-16	Rob.↓	0.21	0.23	0.20	0.18	0.131	0.121
	EAO↑	0.434	0.442	0.464	0.430	0.493	0.535

Table 1. Results on VOT2016. Top-3 results of each dimension (row) are colored in red, blue and green, respectively.

Trackers		DiMP-50 [1]	SiamBAN [4]	D3S [27]	Ocean-off [54]	DCFST [56]	Ours (SAMN)
	Acc.↑	0.590	0.597	0.597	0.64	0.598	0.652
VOT-18	Rob.↓	0.203	0.152	0.178	0.150	0.169	0.145
	EAO↑	0.440	0.452	0.489	0.467	0.452	0.521

Table 2. Results on VOT2018. Top-3 results of each dimension (row) are colored in red, blue and green, respectively.

Trackers		SiamRPN++	ATOM	Retina-MAML [42]	SiamFCOT [19]	Ocean-off	Ours (SAMN)
	Acc.↑	0.580	0.603	0.570	0.601	0.590	0.639
VOT-19	Rob.↓	0.446	0.411	0.366	0.386	0.376	0.231
	EAO↑	0.292	0.292	0.313	0.350	0.327	0.408

Table 3. Results on VOT2019. Top-3 results of each dimension (row) are colored in red, blue and green, respectively.

After training the encoder block, we use Youtube-VOS [48] and freeze the gradients of the encoder to train the decoder. We randomly sample 3 temporally ordered frames from the same video sequence and apply recurrent training strategy. The first frame and its mask are fed into memory encoder. The prediction of second frame is then stored in AMB for predicting the third frame. Then, the loss will be accumulated. We use randomly cropped 384×384 patches for training and set the minibatch to 4. We minimize the cross-entropy loss and mask IoU loss using Adam optimizer [16] with a fixed learning rate of 10^{-5}. First-stage training process takes 120 epochs and decoder training takes 40 epochs using four NVIDIA TITAN XP GPUs.

Testing phase. During inference, the sampling interval in appearance memory bank is set to 5. The output of our model is segmentaion map and will be transferred to rotated box for tracking task. If a ground truth bounding box is available, the SAMN follows the initialization procedure proposed in [27].

4.2. Evaluation

Our tracker achieves the state-of-the-art (sota) results on 6 tracking benchmarks and competitive results on 2 VOS benchmarks. We select representative benchmarks based on their prediction output formats and challenging factors. Our tracker shows its sota performance on three aspects: traditional robust tracking, pixel-level tracking and video object segmentation. Our tracker was evaluated on six major short-term tracking benchmarks and compared with sota trackers: VOT2016 [11], VOT2018 [18], VOT2019 [19], GOT-10k [15], TrackingNet [28], VOT2020 [17]. Our tracker is also evaluated on two popular VOS benchmarks DAVIS16 [33] and DAVIS17 [35].

Rotated Bounding Box format VOT datasets are the most challenging and convincing evaluation tools in tracking. VOT2016, VOT2018 and VOT2019 are widely-used benchmarks for visual object tracking. Each of them contains 60 sequences with various challenging factors. The three datasets are annotated with the rotated bounding boxes, and a reset-based methodology is applied for evaluation. For both benchmarks, trackers are measured in terms of accuracy (A), robustness (R), and expected average overlap (EAO).

Trackers		SiamRPN++	ATOM	DiMP-18	DiMP-50	D3S	Ocean-off	Ours (SAMN)
GOT-10K	$SR_{.75}$↑	32.5	40.2	44.6	49.2	46.2	-	52.2
	AO↑	51.8	55.6	57.9	61.1	59.7	59.2	61.5

Table 4. Results on GOT-10K. Top-3 results of each dimension (row) are colored in red, blue and green, respectively.

Trackers		SiamRPN++	ATOM	DiMP-50	Retina-MAML	D3S	Ours (SAMN)
TrackingNet	Prec.↑	69.4	64.8	68.7	-	66.4	69.7
	Norm. Prec.↑	80.0	77.1	80.1	78.6	76.8	79.4
	Succ.↑	73.3	70.3	74.0	69.8	72.8	74.2

Table 5. Results on TrackingNet. Top-3 results of each dimension (row) are colored in red, blue and green, respectively.

Trackers		SiamMask	STM	DET50 [19]	Ocean	D3S	Ours (SAMN)
	Mask	✓	✓	✓	✓	✓	✓
VOT-20	Acc.↑	0.624	0.751	0.679	0.693	0.699	0.720
	Rob.↓	0.648	0.574	0.787	0.754	0.769	0.794
	EAO↑	0.321	0.308	0.441	0.430	0.439	0.461

Table 6. Results on VOT2020. "Mask" denotes that prediction format is mask. Top-3 results of each dimension (row) are colored in red, blue and green, respectively.

Axis-aligned Bounding Box format Tracking We also evaluate our tracker in the axis-aligned bounding box annotated visual tracking benchmarks, i.e., GOT-10K [15] and TrackingNet [28]. Axis-aligned bounding box annotation is widely-used among object detection and tracking benchmarks. GOT-10K is a recent large-scale dataset (10,000 videos in train subset and 180 in both val and test subset) with 1.5 million bounding boxes. Average overlap (AO), success rates at 75% threshold (SR_{75}) and 50% threshold (SR_{50}) are the three ranking metrics. TrackingNet [28] contains 30000 sequences with 14 million dense annotations and 511 sequences in the test set. It covers diverse object classes and scenes, requiring trackers to have both discriminative and generative capabilities.

Pixel-wise Tracking VOT2020 [17] proposed a significant novelty compared to 2019 that the target position was encoded by a segmentation mask. The VOT2020 [17] benchmark introduces a new evaluation methodology for the promising pixel-wise tracking paradigm which requires trackers to robustly track the target while predicting an accurate binary mask. Segmentation-based trackers need to performs well both in segmentation accuracy and challenging scenarios, e.g., fast motion, distractors and blur. Accuracy (A), robustness (R) and expected average overlap (EAO) are three metrics to evaluate trackers.

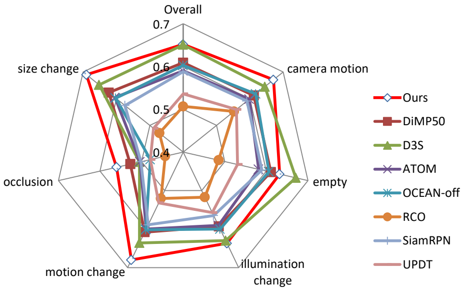

Figure 6. Comparison of accuracy on VOT2018 for the following visual attributes: camera motion, illumination change, occlusion, size change and motion change. Empty means frames do not belong to any of the five attributes.

Video Object Segmentation We also evaluate our tracker in two semi-supervised VOS benchmarks, i.e., DAVIS16&17 [33] [34], following official test protocol: mean Jaccard index (J_M) and mean F-measure (F_M). J_M describes the region similarity while (F_M) measures the contour accuracy of the predictions. VOS datasets typically focus more on segmentation.

4.2.1 Comparing with sota methods

VOT2016&VOT2018&VOT2019: The VOT2016 top performers CCOT [8] and TCNN [29], two recent sota segmentation-based trackers D3S [27] and SiamMask [44], and most recently published sota deep trackers SiamRPN++ [21], SPM [43], UpdateNet [53] and ATOM [7] are compared with our trackers. Table 1 show that our tracker outperforms all trackers on all three measures by a large margin. In terms of EAO, our tracker outperforms the strongest sota tracker D3S by 4.2 points and ATOM by 10.2 points. VOT2018 is the most widely used benchmark so far. We compared SAMN with all official results from [18]. We compared SAMN with the most recent sota trackers: DCFST [56], Ocean [54], D3S [27], SiamBAN [4], DiMP [1], ATOM [7], SiamRPN++ [21] and SiamMask [44]. As shown in Figure, our tracker outperforms all trackers on all three measures by a large margin. In terms of EAO, our tracker outperforms the sota tracker LADCF by 4.2 points and SiamRPN++ by 10.2 points. As shown in Fig. 6, SAMN is more accurate than other trackers towards challenging factors like occlusion, size and motion changes. This shows that our tracker is robust towards occlusion, size changes and motion changes in the target while having the ability to handle with camera motion and illumination changes. SAMN is compared to the recent prevailing trackers. As shown in Table 3, our model surpasses all the competitive trackers in three metrics. Our tracker outperforms the most recent published Siamese correlation tracker Ocean by 2.9 points in EAO. The accuracy of our tracker outperforms the ATOM by 4.6 points. The results demonstrate that our tracking architecture has better performance towards both Siamese correlation trackers and filter-based trackers.

	$\mathcal{J}_{\mathcal{M}}^{16}$	$\mathcal{F}_{\mathcal{M}}^{16}$	$\mathcal{J}\&\mathcal{F}^{16}$	$\mathcal{J}_{\mathcal{M}}^{17}$	$\mathcal{F}_{\mathcal{M}}^{17}$	$\mathcal{J}\&\mathcal{F}^{17}$
Ours(SAMN)	79.0	75.5	77.3	64.8	67.7	66.3
D3S [27]	75.4	72.6	74.0	57.8	63.8	60.8
SiamMask [44]	71.7	67.8	69.8	54.3	58.5	56.4
OnAVOS [40]	86.1	84.9	85.5	61.6	69.1	65.4
STM [31]	84.8	88.1	86.4	69.2	74.0	71.6
MAST [20]	-	-	-	63.3	67.6	65.5
FAVOS [5]	82.4	79.5	80.9	54.6	61.8	58.2
VM [14]	81.0	-	-	56.6	-	-
OSVOS [3]	79.8	80.6	80.2	56.6	63.9	60.3
PLM [38]	75.5	79.3	77.4	-	-	-
OSMN [51]	74.0	72.9	73.5	52.5	57.1	54.8

Table 7. Comparison with segmentation-based trackers and VOS methods on DAVIS16 and DAVIS17.

GOT-10K&TrackingNet: GOT-10K is a recent large-scale dataset consisting of 10K video segments and 1.5 million classical axis-aligned bounding boxes. As shown in Table 4, our tracker improves the SR_{75} by 3.0 points over the sota filter-based tracker DiMP-50, while outperforming DiMP-50 by 0.4 points in terms of AO. Comparing to the Siamese correlation trackers, SAMN outperforms the Ocean by 2.3 points in terms of AO. Compared to the sota tracker D3S, SAMN has improvements of 3.0% on AO and nearly 13.0% improvements on SR_{75}, demonstrating its ability to tracking objects over complex scenes.We further evaluate SAMN on the large-scale TrackingNet. As shown in Table 5, SAMN outperforms the strongest filter-based tracker DiMP-50 by 0.2 points in AUC while our accuracy surpasses the strongest segmentation-based tracker D3S by 3.3 points.

VOT2020. Recently, the tracking community starts focusing on replacing the classical rectangle box with a segmentation mask to accurately represent the target. Our tracker is compared to 6 sota trackers with segmentation ability and 4 trackers with classical bounding box prediction format. All results are from VOT2020 official report [17] or tested by official toolkit. As shown in Table 6, our tracker surpasses all the trackers in EAO measure. SAMN outperforms the top sota tracker DET50 [17] by 1.2 points (0.453 vs. 0.441). Moreover, our tracker significantly outperforms the top sota VOS method STM [30] by 14.5 points in EAO (0.453 vs. 0.308).

DAVIS16&17. Our tracker is compared with the sota segmentation-based trackers and competitive VOS methods. From Table 4.2.1, our tracker outperforms the SOTA segmentation-based trackers D3S and SiamMask [44] by a large margin. On the more challenging benchmark: DAVIS17, our tracker even outperforms all the methods specialized to VOS task except STM in mean of $J\&F$. Compared to D3S, which also belongs discriminative segmentation-based tracker, our approach obtains gains

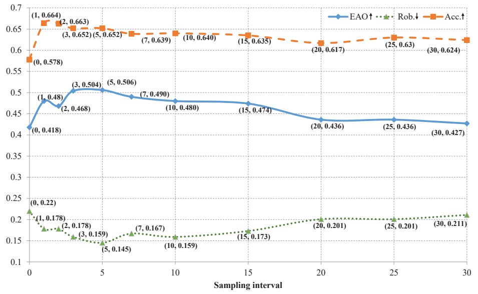

Figure 7. Time interval indicates the sampling interval of memory bank. Zero interval indicates that only the first frame and its ground truth is stored. Up-arrow (down-arrow) indicates higher (lower) is better.

Table 8. Ablation study on VOT2018 and DAVIS16.

Last Add.	✓		✓	✓	✓	✓	✓	✓	✓
Interv.	5	5	5	10	5	5	15	20	5
Filter Samp.	✓	✓	✓	✓		✓			✓
Pos. Encod.	sum	sum	sum	sum	sum	cat	sum	sum	sum
DRM	✓	✓	✓	✓	✓	✓	✓	✓	
A ↑	65.2	66.5	63.5	64.0	62.7	65.0	62.2	62.0	63.4
R ↓	0.145	0.173	0.164	0.159	0.210	0.150	0.225	0.227	0.173
EAO ↑	50.6	46.7	49.2	48.0	42.1	48.6	41.0	40.2	45.1
$\mathcal{J}\&\mathcal{F}^{16}$ ↑	77.3	70.3	-	67.8	69.1	-	66.4	63.6	67.2

of 3.3/5.5 points on J &F for DAVIS16/17, respectively. It demonstrates the strong accurate segmentation ability of our approach.

4.3. Ablation Study

To further show our contributions, we conduct comprehensive ablation studies on VOT2018 and DAVIS16. The performance on tracking and VOS benchmarks can address the robust tracking and accurate segmentation ability of our tracker, respectively.

Temporal Information: We set different sampling interval of AMN. When sampling interval is 0, our tracker is the same as template-matching methods where only the first frame is used. As shown in Fig. 7, the all three measures drop by a large margin in comparison to the modes utilizing temporal information. No temporal information used causes 6.2 points performance drop in EAO. in contrast to storing every sample. It further validates the superiority of our tracking architecture to the template-based trackers.

The amount of samples stored also matters. When sampling interval is 1, our trackers reaches the top accuracy performance which is 0.663. Performance of EAO reaches the top which is 0.506 when sampling interval is 5 frames. Comparing to the 5 frames interval, 30 frames interval which is sparse reduces the EAO by 7.9 points. When the last frame always be added to appearance memory bank, our tracker boosts its overall performance EAO by 3.9 points and robustness performance by 2.8 points when sampling interval is 5 frames.

Positional Encoding: Inspired by CoordConv [26], we concatenate two coordinate channels to read-out features. On the other hand, we simply do positional encoding as that in natural language processing. We add the single spatial matrix to the read-out features. As shown in Tab. 8, adding spatial matrix to the read-out features outperforms the concatenating way by 2 points in terms of EAO. Thus, we choose adding style as our positional encoding way for its simplicity and effectiveness.

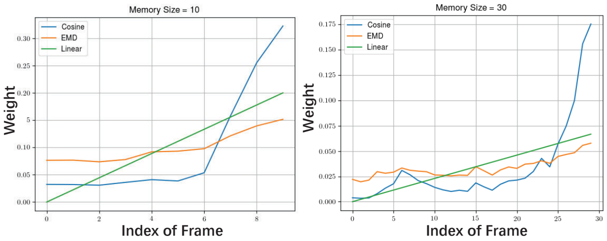

Figure 8. Comparisons of weights generation method.

Sample Filtering: As shown in Table 8, the collaboration between dual memory banks is significant to the overall performance. Without the samples filtered from SMB, the EAO drops from 0.506 to 0.421 when sampling interval equals to 5. The mean of $J\&F$ on DAVIS16 also reduces from 77.3 to 69.1. This indicates that one single memory bank cannot handle these challenging tracking scenarios separately. SAMN can handle with both VOT and VOS tasks while keeping fast inference speed.

Validation of the DRM To evaluate the contribution of the EMD to reweight samples in temporal domain on tracking task, we compare it with cosine distance and linear decay strategy. The weights generations are compared in Fig. 8. The cosine metric significantly assigns larger values to recent frames which eliminates the impacts of past frames. EMD metrics can balance the weight values between cosine metric and linear time decay strategy. Table 8 also validates the improvements from DRM.

5. Conclusion

In this work, we propose a new tracking architecture which fully exploits temporal information by spatio-appearance memory network. Our tracker shows its sota performance on three prediction format. This fully demonstrates the promising potential of segmentation-based tracking methods. In the future, we will further improve the SAMN tracking architecture, especially in efficient memory management and make dual memory networks more collaborative and unified. We hope to develop a model that has sota performance on both VOT and VOS tasks while keeping real-time inference speed under this architecture.

Acknowledgment

This work was supported by the NSFC under Nos. 61773117, 61876088 and 62006041.

References

[1] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019.

[2] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010.

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, pages 221–230, 2017.

[4] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, 2020.

[5] 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 *CVPR*, pages 7415–7424, 2018.

[6] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 2014.

[7] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019.

[8] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*. Springer, 2016.

[9] 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 *CVPR*, 2019.

[10] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[11] SJ Hadfield, R Bowden, and K Lebeda. The visual object tracking vot2016 challenge results. *Lecture Notes in Computer Science*, 9914:777–823, 2016.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018.

[14] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, pages 54–70, 2018.

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

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Martin Danelljan, Alan Lukezic, Ondrej Drbohlav, Linbo He, Yushan Zhang, Song Yan, Jinyu Yang, Gustavo Fernandez, and et al. The eighth visual object tracking vot2020 challenge results, 2020.

[18] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka ˇCehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *ICCV*, 2018.

[19] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, Abdelrahman Eldesokey, Jani Kapyla, and Gustavo Fernandez. The seventh visual object tracking vot2019 challenge results, 2019.

[20] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, 2020.

[21] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

[22] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[23] Peihua Li. Tensor-sift based earth mover's distance for contour tracking. *Journal of mathematical imaging and vision*, 46(1):44–65, 2013.

[24] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.

[25] 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 *ECCV*, 2014.

[26] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, 2018.

[27] Alan Lukezic, Jiri Matas, and Matej Kristan. D3s-a discriminative single shot segmentation tracker. In *CVPR*, 2020.

[28] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.

[29] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242*, 2016.

[30] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[31] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9226–9235, 2019.

[32] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017.

[33] 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 *CVPR*, 2016.

[34] 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 *CVPR*, pages 724–732, 2016.

[35] 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.

[36] Suyash Shetty. Application of convolutional neural network for image classification on pascal voc challenge 2012 dataset. *arXiv preprint arXiv:1607.03785*, 2016.

[37] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 2015.

[38] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, pages 2167–2176, 2017.

[39] Chong Sun, Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Correlation tracking via joint discrimination and reliability learning. In *CVPR*, 2018.

[40] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, pages 1000–1008, 2017.

[41] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[42] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. *ArXiv*, 2020.

[43] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Spm-tracker: Series-parallel matching for real-time visual object tracking. In *CVPR*, 2019.

[44] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.

[45] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[46] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *ICLR*, 2015.

[47] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013.

[48] 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 *ECCV*, 2018.

[49] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[50] Yinda Xu, Zeyu Wang, xin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, 2020.

[51] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, pages 6499–6507, 2018.

[52] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *ECCV*, 2018.

[53] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, 2019.

[54] Zhipeng Zhang and Houwen Peng. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020.

[55] Qi Zhao, Zhi Yang, and Hai Tao. Differential earth mover's distance with its applications to visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):274–287, 2008.

[56] Linyu Zheng, Ming Tang, Yingying Chen, Jinqiao Wang, and Hanqing Lu. Learning feature embeddings for discriminant model based tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.