Learning to Track Objects from Unlabeled Videos

Jilai Zheng[1] Chao Ma[1*] Houwen Peng[2] Xiaokang Yang[1]
[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2] Microsoft Research
{zhengjilai,chaoma,xkyang}@sjtu.edu.cn, houwen.peng@microsoft.com

Abstract

In this paper, we propose to learn an Unsupervised Single Object Tracker (USOT) from scratch. We identify that three major challenges, i.e., moving object discovery, rich temporal variation exploitation, and online update, are the central causes of the performance bottleneck of existing unsupervised trackers. To narrow the gap between unsupervised trackers and supervised counterparts, we propose an effective unsupervised learning approach composed of three stages. First, we sample sequentially moving objects with unsupervised optical flow and dynamic programming, instead of random cropping. Second, we train a naive Siamese tracker from scratch using single-frame pairs. Third, we continue training the tracker with a novel cycle memory learning scheme, which is conducted in longer temporal spans and also enables our tracker to update online. Extensive experiments show that the proposed USOT learned from unlabeled videos performs well over the state-of-the-art unsupervised trackers by large margins, and on par with recent supervised deep trackers. Code is available at https://github.com/VISION-SJTU/USOT.

1. Introduction

Visual object tracking is one of the most fundamental computer vision tasks with numerous applications, such as autonomous driving, intelligent surveillance, robot, and human-computer interaction. The past few years have witnessed considerable progress in visual object tracking, thanks to the powerful representation of deep learning. In spite of the success, the state-of-the-art deep tracking algorithms are data-hungry, requiring a huge number of annotated data for supervised training. As manually labeled data are expensive and time-consuming, exploiting ubiquitous unlabeled videos for visual tracking has drawn increasing attention recently. Following the classic pipeline of unsupervised learning, existing unsupervised trackers randomly crop template regions on unlabeled videos and employ either self consistency [34] or cycle consistency [37] as a pretext task for learning to track. Despite the promising results, there still exists a large performance gap between unsupervised and supervised trackers. In view of the great success of unsupervised learning on a number of other vision tasks, such as video object segmentation [23], optical flow [28] and depth estimation [14], it is of great interest to narrow the gap between unsupervised and supervised trackers.

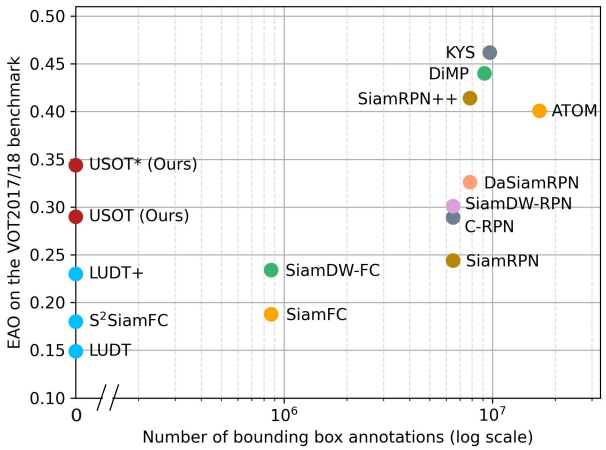

Figure 1: Comparison on the VOT2017/18 benchmark with recent deep trackers. The proposed trackers, USOT and USOT*, perform well over the state-of-the-art unsupervised deep trackers, including LUDT [38], LUDT+ [38], and S^{2}SiamFC [34], and on par with recent supervised trackers. Notation: SiamFC [2], SiamDW [47], SiamRPN [26], C-RPN [12], DaSiamRPN [50], ATOM [8], SiamRPN++ [25], DiMP [3], KYS [4].

We identify three critical challenges that cause the performance bottleneck of unsupervised trackers. *1) Moving object discovery.* As ground truth bounding boxes are not available, existing unsupervised trackers randomly sample regions in frames as pseudo templates [37, 34]. Random samples are far from precisely locating objects, not to mention learning to distinguish between objects and background. Moreover, as random samples do not contain clear edges of objects, they are not suitable for bounding box re-

* Corresponding author.

gression learning. The lack of bounding box regression for scale change estimation heavily limits the performance of unsupervised trackers. *2) Rich temporal variation exploitation.* Due to the lack of labels in the temporal span, existing unsupervised trackers struggle to learn from rich motion clues. For example, UDT [37] performs forward and backward tracking within at most 10 frames. In such a short clip, the foreground objects show highly correlated appearances with little variations, causing a failure to exploit rich temporal variations over a long span for training. *3) Online update.* Online update helps to exploit the temporal smoothness and has demonstrated great success in leading supervised tracking methods [35, 3, 46, 48]. While supervised trackers usually collect multiple object samples in separated frames for learning online modules [3, 13], it is more challenging to train online branches for unsupervised trackers, due to lack of even coarse object locations in videos.

To address these challenges, we propose to train a robust tracker from unlabeled videos. First, for data preparation, we develop a sequential box sampling algorithm to coarsely discover moving objects from unlabeled videos. Specifically, we use unsupervised optical flow to detect moving objects and apply dynamic programming to sequentially link candidate boxes. Second, we naively train from scratch an unsupervised Siamese tracker using single-frame pairs. That is, we train with each Siamese pair cropped based on the sampled box in a single frame. Despite its simplicity, we show that this strategy provides a great initialization for the unsupervised tracker, thus beneficial to future training in longer temporal spans. Third, we propose a cycle memory learning scheme to continue training the naive tracker. Specifically, we divide the whole video into a number of fragments according to the detected moving object trajectory. We then conduct forward tracking from a single frame to several other frames in the same fragment, and store all intermediate tracking results in a memory queue. We then track backward to the initial frame to compute the consistency loss. Since the length of video fragments is quite long (averaged 64.6 frames on VID [33]) compared with UDT [37] ($<$10 frames), our tracker can capture large motion and appearance variations. More importantly, the proposed cycle memory scheme allows updating the memory queue online for model update (see Sec. 4.3).

We evaluate the proposed unsupervised tracker on six large-scale benchmarks. Extensive experiments show that our proposed tracker performs well over the state-of-the-art unsupervised trackers by large margins, and on par with the recent supervised trackers (see Fig. 1). The main contributions of this work are summarized as follows:

- We coarsely discover moving objects from unlabeled videos for unsupervised learning.
- We train a naive Siamese tracker with single-frame pairs and gradually extend it to longer temporal spans.
- We propose a cycle memory learning scheme, allowing unsupervised trackers to update online.

2. Related Work

Supervised Tracking. Deep learning has revolutionized the field of visual tracking by powerful representation. In the past few years, template-based deep trackers with Siamese networks have received increasing attention due to the promising results on benchmark datasets. These trackers regard the target object as a template and search over a cropped window to locate the target. SiamFC [2] first utilizes the same backbone network to extract deep features from both the template patch and the search patch, and computes the response map with cross-correlation. Since then, considerable efforts have been made to extend Siamese trackers. SiamRPN [26] incorporates a region proposal network (RPN) [31] into the Siamese framework. DiMP [3] proposes to attach an online module to Siamese trackers for template update. Other noticeable improvements in the Siamese framework involve advanced backbone network [47], correlation method [25], attention mechanism [45], re-detection module [36], mask generation [40], feature alignment [48], anchor-free regressor [6, 15], etc. With these efforts, Siamese trackers have shown the superior tracking performance thus far. However, training Siamese trackers requires a huge number of labeled training data. In this work, we aim at a novel unsupervised learning scheme, which helps to learn template-based trackers from unlabeled videos in the wild.

Unsupervised Tracking. The pioneer unsupervised deep tracker (UDT) [37] suggests that a robust tracker is able to track an object forward and backward in a video and finally return to its initial location in the starting frame. UDT develops a tracker based on DCFNet [39], and computes a cycle consistency loss between forward and backward trajectories in the training phase. The contemporary method of UDT is TimeCycle [41], which proposes cycle consistency to generate unsupervised video representation. JSLTC [27] proposes to calculate an inter-frame affinity matrix to model the transitions between video frames and use such correspondence to track objects. S^2SiamFC [34] adopts the self-Siamese pipeline to train a foreground/background classifier like SiamFC [2] from single-frame pairs, showing comparable results to the supervised counterpart. Despite the promising results, there exists a large performance gap between the state-of-the-art unsupervised trackers and the top-performing supervised ones. We identify three critical challenges, moving object discovery, rich temporal variation exploitation, and online update, that cause the performance bottleneck of unsupervised trackers. By effectively tackling these challenges, the proposed unsupervised tracker outperforms the state-of-the-art unsupervised trackers by large margins, and is on par with recent supervised trackers.

3. Proposed Method

In this section, we present the proposed unsupervised tracker in detail. The unsupervised training scheme involves three stages. The first stage in Sec. 3.1 aims to produce a trajectory of moving objects from unlabeled videos. The second stage in Sec. 3.2 learns a naive Siamese tracker using single-frame pairs. The third stage in Sec. 3.3 continues training the naive tracker by means of cycle memory learning, which is performed in longer temporal spans and also enables the unsupervised tracker to update online.

3.1. Moving Object Discovery

Instead of randomly cropping objects, we propose to generate a smooth bounding box sequence for moving foreground objects in unlabeled videos. For discovering moving objects, we have two key observations:

- Foreground objects tend to have distinguishing motion patterns in contrast to the background surroundings. This inspires us to discover candidate foreground objects by means of unsupervised optical flow.
- The trajectories of moving objects tend to be smooth. This motivates us to employ dynamic programming (DP) to get temporally reliable box sequences.

Candidate Box Generation. Let an arbitrary video be $\mathcal{I}$ including L successive frames with the same size $W \times H$, namely $\mathcal{I} = \{I_t \mid 1 \leq t \leq L\}$, where I_t is the t^{th} frame in $\mathcal{I}$. To locate the potential foreground object in frame I_t, we first compute the optical flow map F_t with frame I_t and frame I_{t+T_f}, i.e., $F_t = Flow_{t \to t+T_f}$. T_f is an interval for computing optical flow. As is illustrated in Fig. 2, we obtain F_t from frame I_t and frame I_{t+T_f} using the off-the-shelf unsupervised ARFlow [28] algorithm, and then transform F_t to a distance map D_t. We binarize D_t to get a mask M_t as follows:

$$M_t^i = \begin{cases} 1 & \text{if } D_t^i \geq \alpha \cdot \max_j(D_t^j) + (1-\alpha) \cdot \text{mean}_j(D_t^j) \\ 0 & \text{o.w.} \end{cases}$$
$$\text{where } D_t^i = \left\| F_t^i - \text{mean}_j(F_t^j) \right\|_2, \tag{1}$$

and $\alpha \in (0, 1)$ is a hyper-parameter. Here superscript denotes pixel-wise index. The maximum value and mean value within the spatial dimension are respectively indicated by $\max$ and mean.

Every connected area with all internal pixels satisfying $M_t^i = 1$ corresponds to an area which has distinguishing motion compared with background in I_t, and this area is more likely to cover a foreground object. To further filter out unreliable areas from these candidates, we take the circumscribed rectangles of all these areas as initial candidate boxes, and score the boxes according to their sizes and positions. Due to center bias, larger bounding boxes in the middle of the image should have higher quality scores. Let $B = (x_0, y_0, x_1, y_1)$ denote the top-left and bottom-right corners of a box. The quality score S_c of the box B is defined as:

$$\begin{aligned} S_c(B) =& (x_1 - x_0)(y_1 - y_0) + \\ & \beta \cdot \min(x_0, W - x_1) \min(y_0, H - y_1), \end{aligned} \tag{2}$$

where β is a weight parameter. The box with the highest score is selected as the final candidate box B_t for frame I_t. We denote the set of all these selected candidate boxes in video $\mathcal{I}$ as $\mathcal{B} = \{B_t \mid 1 \leq t \leq L\}$.

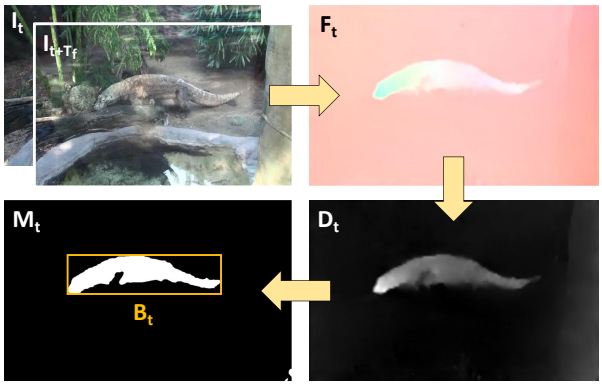

Figure 2: Candidate box generation via optical flow. The flow map F_t contains the distinguishing motion patterns of moving objects. We binarize the flow map F_t using a distance metric D_t to generate the candidate box B_t.

Box Sequence Generation. The generated candidate bounding boxes $\mathcal{B}$ may contain noisy boxes due to camera shake, occlusion, etc. To remove unreliable boxes, we apply dynamic programming to create a more reliable bounding box sequence $\mathcal{B}'$. According to the second observation that the trajectory of a moving object in a video should be smooth, we select a subset of candidate bounding boxes from $\mathcal{B}$, where the trajectory of the selected boxes is as smooth as possible. For dynamic programming, the most critical issue is how to measure the reward of transition in the box trajectory from one bounding box to another. We modify the DIoU [49] metric, which originally considers the overlap and distance between two boxes. Formally, the reward R_{dp} for dynamic programming is defined as:

$$R_{dp}(B_t, B_{t'}) = IoU(B_t, B_{t'}) - \gamma \cdot R_{DIoU}(B_t, B_{t'}), \tag{3}$$

where γ is a hyper-parameter. To encourage a smooth trajectory, we set $\gamma > 1$ for the distance penalty on R_{DIoU} [49]. Note that dynamic programming aims at discovering an optimal path in the box sequence $\mathcal{B}$ with the highest reward accumulation (see the supplementary document for the complete algorithm). As shown in Fig. 3, for the frames whose candidate boxes are not selected by DP, we use linear interpolation to generate pseudo boxes based on their adjacent candidate boxes selected by DP.

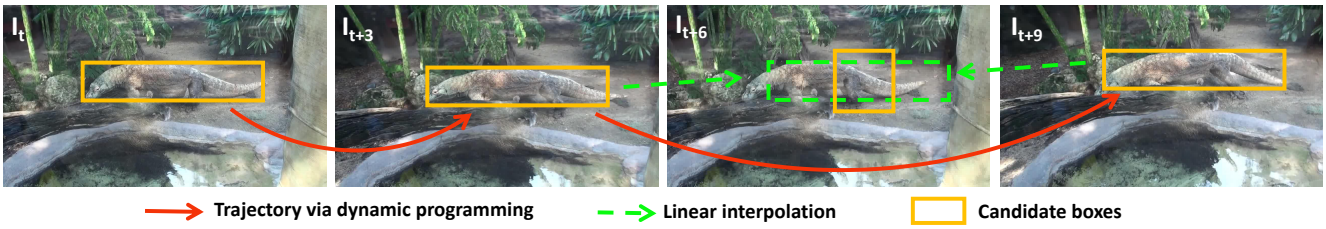

Figure 3: Box sequence generation. We use dynamic programming to generate a smooth and reliable box trajectory from candidate boxes in yellow. Pseudo boxes in green in the remaining frames are generated through linear interpolation.

3.2. Naive Siamese Tracker

With the generated box sequences, we train a naive Siamese tracker using single-frame pairs from scratch. This pretext task is based on a simple observation that an image and any of its sub-region naturally form a training pair of the Siamese network [34]. However, randomly sampled pseudo boxes as in [34] fail to cover foreground objects for effectively training Siamese networks. Moreover, random samples are not suitable for learning bounding box regression. This significantly hinders the performance of unsupervised trackers. We propose to utilize the reliable box sequence $\mathcal{B}'$ as training data. To ensure that the most precise bounding boxes in $\mathcal{B}'$ are sampled by the data loader, we adopt a two-level scoring mechanism to filter out low-quality boxes at both the sequence and frame levels. We find that denser frame selection by DP in video $\mathcal{I}$ tends to imply a more successful moving object discovery. As such, we define the quality score Q_v of video $\mathcal{I} = \{I_t \mid 1 \leq t \leq L\}$ as:

$$Q_v(\mathcal{I}) = \frac{N_{dp}}{L}, \tag{4}$$

where N_{dp} indicates the number of frames in video $\mathcal{I}$ selected by DP.

Similarly, the frame quality score evaluating the box B'_t in frame I_t can be measured by the percentage of frames selected by DP within all its adjacent frames. Let T_s be a fixed frame interval. We formally define the frame quality score Q_f as:

$$Q_f(B'_t) = \frac{N'_{dp}}{2T_s + 1}, \tag{5}$$

where N'_{dp} indicates the number of frames between frame I_{t-T_s} and frame I_{t+T_s} selected by DP.

When loading training pairs, we sequentially conduct video sampling and frame sampling. We sample a video only if its quality score $Q_v(\mathcal{I}) \geq \theta_1$, where θ_1 is a threshold. During frame sampling, we randomly sample several frames with their total number positively correlated with $1/Q_v(\mathcal{I})$ from the selected video, and then select the frame with the highest frame quality score $Q_f(B'_t)$ for training.

We follow the conventional training paradigm as in SiamFC [2]. The input template z_t and the search area x_t are respectively of size 127×127 and 255×255, both cropped from I_t based on B'_t. After extracting deep features from the input pair, we adopt PrPool [19] to pool the template feature, and then compute the multi-scale correlation map [48]. The output response map $\mathcal{R}_{cls}$ is of size $25 \times 25 \times 1$ for foreground/background classification. The other output response map $\mathcal{R}_{reg}$ is of size $25 \times 25 \times 4$ for regressing the distances from the center location to the four sides of the bounding box. The loss function $\mathcal{L}_{naive}$ is the sum of both the regression and classification losses:

$$\mathcal{L}_{naive} = \mathcal{L}_{reg} + \lambda_1 \mathcal{L}_{cls}, \tag{6}$$

where $\mathcal{L}_{reg}$ and $\mathcal{L}_{cls}$ are respectively the IoU loss [44] and the binary cross-entropy (BCE) loss [10]. λ_1 is a weight parameter.

3.3. Cycle Memory Training

We view the above unsupervised Siamese tracker as a naive tracker as it incurs two limitations. First, as the template and search area are cropped in the same frame, the tracker is not learned with large motion and appearance variations. Second, this tracker cannot update itself online, thus fails to track objects in long temporal spans or under complex scenes.

We propose to continue training the naive tracker using a cycle memory learning scheme, aiming to enable the tracker to handle large variations as well as update the memory queue online. The main idea of cycle memory can be summarized as in Fig. 4. In brief, we first conduct forward tracking from a template z_t to N_{mem} adjacent memory frames, then store features of all intermediate tracking results as a memory queue, and finally conduct backward tracking to the original search area x_t. A cycle memory loss $\mathcal{L}_{mem}$ is computed using the same ground truth as $\mathcal{L}_{cls}$.

Specifically, at every training step, we simultaneously crop a training pair z_t and x_t in frame I_t (the same as training the naive Siamese tracker), as well as N_{mem} memory search areas sampled from $\{x_t \mid T_l \leq t \leq T_u\}$. These memory search areas are cropped from N_{mem} adjacent frames of I_t according to the box sequence $\{B'_t \mid T_l \leq t \leq T_u\}$. Here T_l and T_u are the lower and upper frame indices for sampling memory frames. Selecting these two indices is quite important. To learn from long-term variations, the frame interval between T_l and T_u

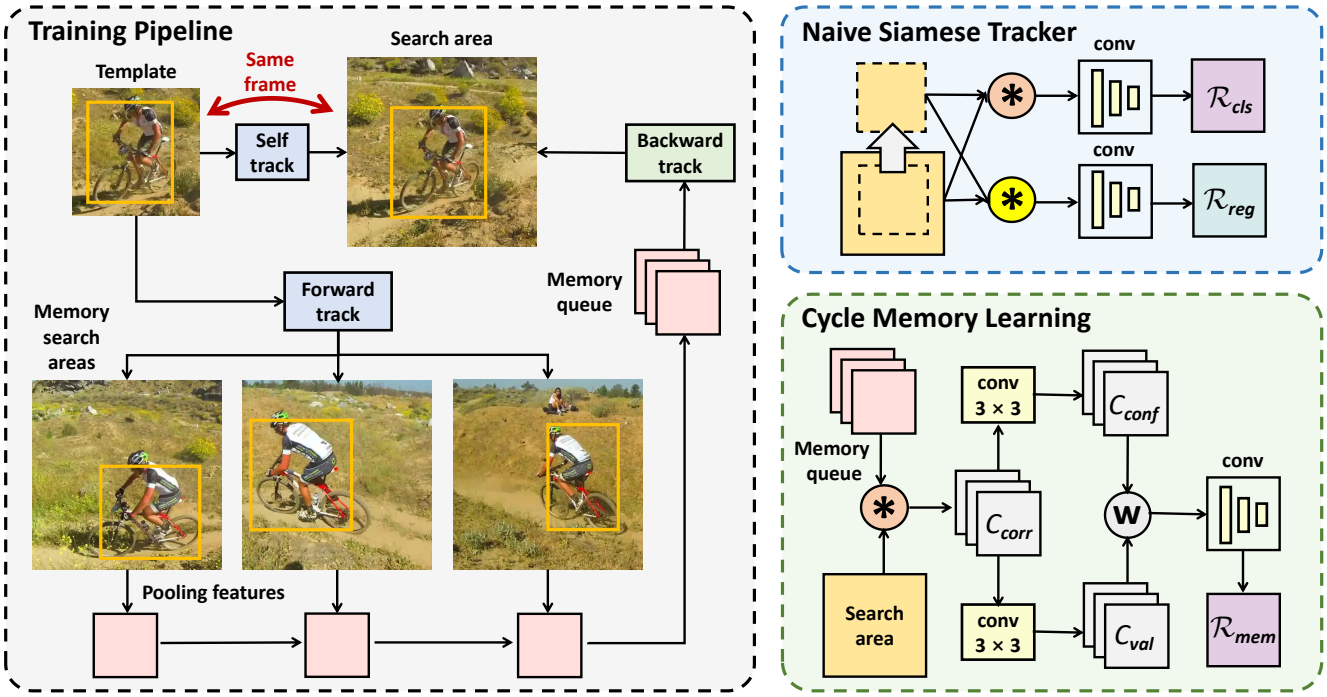

Figure 4: Overview of the proposed unsupervised tracking framework. Left: The overall training pipeline. Right: The detailed structure of the naive Siamese tracker for *self tracking* and *forward tracking*, and the online module learned with the cycle memory scheme. The naive tracker is trained with a template and a search area cropped from the same frame, while the online module aims to *track backward* from the memory search areas to the template frame following the cycle learning pipeline. The circle notations with $*$ denote multi-scale correlation [48] for deep features, where the same color indicates weight sharing. The circle with W refers to the confidence-value module for integrating the correlation maps (Eqn. 8).

should be large enough. However, excessive frame interval does harm to the learning process as the target object may disappear in frames far from I_t. In practice, we dynamically set T_l and T_u at frame I_t. Since they are two mirror variables, we formally define T_u as follows:

$$\begin{aligned} T_u(I_t) = & \max_{t \le k \le L} \{k\} \\ \text{s.t.} \quad & \forall t < t' \le k, R_{dp}(B'_{t'-1}, B'_{t'}) \ge \theta_2 \\ & \forall t < t' \le k, Q_f(B'_{t'}) \ge \theta_3, \end{aligned} \tag{7}$$

where θ_2 and θ_3 are two thresholds. The main idea of Eqn. 7 is that, as long as a box B'_k can be connected to B'_t through a smooth and reliable box sequence in $\mathcal{B}'$, search area cropped from B'_k can be used for cycle memory training. In other words, we take step changes in $\mathcal{B}'$ to divide $\mathcal{I}$ into fragments, and the pseudo boxes of all frames in the same fragment tend to locate the same object. This scheme helps our tracker exploit long-term variations, while still ensuring the reliability of pseudo bounding boxes in the memory frames (see Sec. 4.3 for quantitative analysis).

Let N_{mem} denote the number of memory frames. We first utilize the tracker to predict N_{mem} intermediate bounding boxes in the memory frames for the template z_t. We adopt PrPool [19] to pool N_{mem} features based on the intermediate boxes. Then we use the pooled features as templates to conduct multi-scale correlation [48] with the deep feature of x_t. Note that the original classification branch and the memory branch share the same weights in terms of the multi-scale correlation module. All N_{mem} correlation maps, denoted as $\{C^u_{corr} \mid 1 \le u \le N_{mem}\}$, are integrated together by a confidence-value strategy. Specifically, given a correlation map C^u_{corr}, we utilize two 3×3 convolution layers to generate a confidence map C^u_{conf} and a value map C^u_{val} with the same dimension. We then normalize C^u_{conf} element-wise across all confidence maps as weights on C^u_{val}. The finally integrated correlation map C is formulated as follows:

$$C = \sum_{1 \le u \le N_{mem}} \text{softmax}\left(C^u_{conf}\right) \odot C^u_{val}, \tag{8}$$

where $\odot$ denotes Hadamard product. As shown in Fig. 4, the integrated map C is converted to $25 \times 25 \times 1$ via convolution, yielding the response map $\mathcal{R}_{mem}$ of the object in search area x_t. The total loss function $\mathcal{L}$ for training is:

$$\mathcal{L} = \mathcal{L}_{reg} + \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{mem}, \tag{9}$$

where λ_1 and λ_2 are weight parameters. We use the BCE loss [10] as the cycle memory loss $\mathcal{L}_{mem}$, which shares the same pseudo ground truth label as in $\mathcal{L}_{cls}$.

Table 1: Results on the VOT benchmarks. The proposed trackers perform well over the state-of-the-art unsupervised trackers. Boldface denotes the best performance among all trackers built without video labels for offline training (the same below).

Tracker	Unsup.	VOT2016			VOT2017/18			VOT2020		
		A ↑	R ↓	EAO ↑	A ↑	R ↓	EAO ↑	A ↑	R ↑	EAO ↑
SiamFC [2]	No	0.532	0.461	0.235	0.503	0.585	0.188	0.418	0.502	0.179
DaSiamRPN [50]	No	0.61	0.22	0.411	0.56	0.34	0.326	-	-	-
ATOM [8]	No	-	-	-	0.590	0.204	0.401	0.462	0.734	0.271
DiMP [3]	No	-	-	-	0.597	0.153	0.440	0.457	0.740	0.274
IVT [32]	Yes	0.419	1.109	0.115	0.400	1.639	0.076	0.345	0.244	0.092
MIL [1]	Yes	0.407	0.727	0.165	0.394	1.011	0.118	0.367	0.322	0.113
KCF [17]	Yes	0.489	0.569	0.192	0.447	0.773	0.135	0.407	0.432	0.154
ECO [7]	Yes	0.55	**0.20**	0.375	0.48	**0.27**	0.280	-	-	-
S^2SiamFC [34]	Yes	0.493	0.639	0.215	0.463	0.782	0.180	-	-	-
LUDT [38]	Yes	0.544	0.422	0.232	0.463	0.693	0.154	-	-	-
LUDT+ [38]	Yes	0.570	0.331	0.299	0.490	0.412	0.230	-	-	-
USOT (Ours)	Yes	0.593	0.336	0.351	0.564	0.435	0.290	**0.458**	**0.600**	**0.222**
USOT* (Ours)	Yes	**0.600**	0.233	**0.402**	**0.578**	0.304	**0.344**	0.448	**0.600**	0.219

4. Experiments

This section presents the results of our unsupervised tracker on multiple benchmarks, with comparisons to the state-of-the-art tracking algorithms. Extensive ablation studies are provided to analyze the effectiveness of our design choices.

4.1. Implementation Details

Training. Our tracker is trained on the data collected from the training sets of four datasets including GOT-10k [18], ImageNet VID [33], LaSOT [11] and YouTube-VOS [43]. Note that the ground-truth labels of these training sets are not available in our method. The hyper-parameters for extracting the video box sequences are set as $\alpha = 0.3$, $\beta = 0.5, \gamma = 4.1$ respectively. Our network adopts ResNet-50 [16] as the backbone network, and uses the third convolutional block to extract deep features for input images. We notice that existing CNN backbones pretrained on the ImageNet dataset [33] contain information from manual labels. For the sake of solid comparisons, we conduct all the experiments in two settings (i.e., w/o and w/ supervised backbone pretraining on ImageNet). During training, we use synchronized SGD [24] on 4 NVIDIA GeForce RTX 3090 GPUs. Each GPU hosts 12 groups of training instances. The whole end-to-end training phase takes 30 epochs in total, in which cycle memory is conducted only within the last 25 epochs. We start with a warm-up learning rate from 2.5×10^{-3} to 5×10^{-3} in the first 6 epochs, while the remaining epochs adopt an exponentially decreasing learning rate from 5×10^{-3} down to 2×10^{-5}.

At each training step, we sample a template bounding box from $\mathcal{B}'$ with $\theta_1 = 0.4$ and crop a template patch and a search area as SiamFC [2]. This image pair is augmented with horizontal and vertical flips. For training the online module with cycle memory, we input extra $N_{mem} = 4$ memory search areas together with the template-search pair. Other hyper-parameters for cycle memory are set to $\gamma = 2.5$, $\theta_2 = 0.45$ and $\theta_3 = 0.40$. The trade-off parameter in the loss function is fixed to $\lambda_1 = 0.2$ for naive Siamese training, and for cycle memory we keep $\lambda_1 + \lambda_2 = 0.9$ and gradually decrease λ_1 from 0.3, in order to make the tracker gently suit to the tracking task in longer temporal spans.

Inference. The inference is performed on both the offline and online branches. The offline branch follows the conventional inference methodology of Siamese networks [25, 48]. Based on the template feature from frame I_1, the offline response maps $\mathcal{R}_{cls}$ and bounding boxes $\mathcal{R}_{reg}$ in subsequent frames are generated by the offline classifier and regressor. On the other hand, the online branch maintains a memory queue of templates and dynamically updates this queue with new predictions. In practice, when tracking the object in frame I_t, the memory queue consists of totally N_q templates, including two ground-truth templates from frame I_1 (i.e., the original template and its horizontal flip), the latest predicted template from frame I_{t-1}, and $N_q - 3$ historical templates with the highest scores in $\mathcal{R}$ from frame I_2 to I_{t-2}. The final response map $\mathcal{R}$ is a weighted sum of $\mathcal{R}_{cls}$ and $\mathcal{R}_{mem}$, namely $\mathcal{R} = (1-w)\mathcal{R}_{cls} + w\mathcal{R}_{mem}$, where the weight w is set to 0.7 throughout our experiments.

4.2. State-of-the-art Comparison

We compare our method with the state-of-the-art unsupervised and supervised trackers. The comparisons are conducted on six benchmarks, including VOT2016 [20], VOT2017/18 [22], VOT2020 [21], TrackingNet [29], OTB2015 [42] and LaSOT [11]. We denote by USOT the

Table 2: Results on the TrackingNet [29] dataset. The proposed unsupervised trackers, USOT and USOT*, perform well over the state-of-the-arts.

Trackers	Unsup.	Succ. ↑	Prec. ↑	Norm P. ↑
SiamFC [2]	No	57.1	53.3	66.3
MDNet [30]	No	61.4	55.5	71.0
UpdateNet [46]	No	67.7	62.5	75.2
ATOM [8]	No	70.3	64.8	77.1
SiamRPN++ [25]	No	73.3	69.4	80.0
KCF [17]	Yes	44.7	41.9	54.6
DSST [9]	Yes	46.4	46.0	58.8
ECO [7]	Yes	56.1	48.9	62.1
LUDT [38]	Yes	54.3	46.9	59.3
LUDT+ [38]	Yes	56.3	49.5	63.3
USOT (Ours)	Yes	59.9	55.1	68.2
USOT* (Ours)	Yes	**61.5**	**56.6**	**69.1**

proposed tracker trained with the unsupervised backbone initialization [5], while denoting by USOT* that with supervised ImageNet pretraining. It is worth mentioning that we keep all the hyper-parameters fixed throughout the experiments to report results on all the datasets.

VOT2016. The VOT2016 dataset contains 60 video sequences. We adopt the Accuracy (A), Robustness (R) and Expected Average Overlap (EAO) [20] as the VOT-toolkit to evaluate the overall performance. Tab. 1 shows that the proposed trackers, both USOT and USOT*, significantly outperform the state-of-the-art unsupervised trackers, with respectively 5.2 and 10.3 points increase in EAO over the top-performing unsupervised tracker LUDT+.

VOT2017/18. The VOT2017/18 dataset consists of 60 more challenging video sequences. Tab. 1 shows that our USOT* obtains 8.8 and 11.4 points increase respectively in Accuracy and EAO compared with LUDT+. Our USOT without supervised backbone still outperforms LUDT+ by respectively 7.4 and 6.0 points in Accuracy and EAO.

VOT2020. The VOT2020 dataset encodes targets with segmentation masks, and updates the calculation of Accuracy (A), Robustness (R) and Expected Average Overlap (EAO) [21]. Tab. 1 shows that our USOT* outperforms SiamFC by 3.0, 9.8 and 4.0 points respectively in A, R and EAO. Our USOT even achieves better performance compared with USOT* with an EAO of 0.222.

TrackingNet. The TrackingNet dataset contains over 30000 videos, with 511 videos for testing. Tab. 2 shows that our USOT* outperforms LUDT+ with 5.2 points in Success and 7.1 points in Precision. Our USOT is comparable with USOT* on the TrackingNet dataset. This suggests that unsupervised representation learning has the potential to perform as well as supervised ImageNet pretraining.

OTB2015. The OTB2015 dataset contains 100 videos. Tab. 3 shows that our USOT outperforms USOT*. This

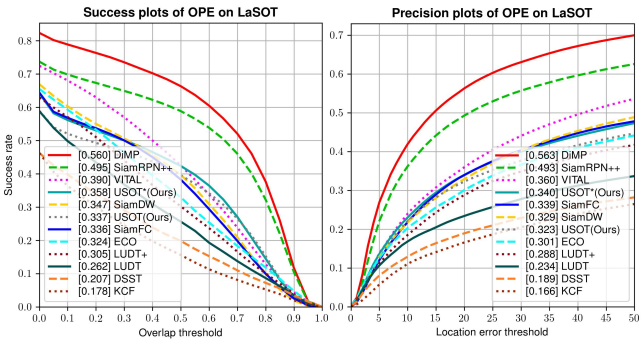

Figure 5: Success plot and Precision plot on the LaSOT testing set [11].

affirms the potential of unsupervised representation learning from scratch. Furthermore, the proposed unsupervised trackers achieve comparable performance with LUDT and SiamFC. Our trackers perform slightly worse than LUDT+, because LUDT+ adopts some online tracking techniques in [7] over LUDT for better performance.

LaSOT. The LaSOT testing set consists of 280 videos with an average length over 2000. LaSOT is important for measuring long-term tracking performance. Fig. 5 shows that our USOT* significantly outperforms LUDT+ with an increase of 5.3 and 5.2 on Success and Precision, respectively. Our USOT achieves comparable results to the supervised trackers SiamFC [2] and SiamDW [47].

4.3. Ablation Studies

Training Stage Indispensability. We do extensive ablation studies on the importance of different stages in the training phase. Experiments are conducted on the VOT2017/18 benchmark with USOT*. Tab. 4 shows that training a naive tracker from single-frame pairs with random cropping causes a significant accuracy drop compared with our proposed box sequence generation. In addition, directly conducting cycle memory training without the naive Siamese tracker initialization also causes a large performance drop.

Video Utilization Rate. We use the video quality score $Q_v(\mathcal{I})$ in Eqn. 4 to filter out noisy sequences during unsupervised training. On the GOT-10k dataset and the VID dataset, we utilize 50.8% and 56.3% videos respectively within all videos available for training, covering rich varieties of unlabeled videos.

Frame Interval. Our proposed training method can learn the appearance information across long intervals. This facilitates unsupervised trackers to adapt to temporal appearance changes. Compared to the very short frame intervals in previous deep unsupervised trackers S^2SiamFC [34] (i.e., 0 frame) and UDT [37] (i.e., < 10 frames), the training instances sampled by our method possess an averaged long frame interval of 41.1 and 64.6 respectively on the GOT-10k and VID datasets.

Table 3: Results on the OTB2015 [42] dataset. Our unsupervised trackers achieve comparable results with the previous supervised and unsupervised trackers.

Tracker	DCFNet [39]	SiamFC [2]	SiamRPN [26]	ATOM [8]	DiMP [3]	DSST [9]	KCF [17]	LUDT [38]	LUDT+ [38]	USOT (Ours)	USOT* (Ours)
AUC success	58.0	58.2	63.7	66.7	68.6	51.8	48.5	60.2	**63.9**	58.9	57.4
Distance precision	-	77.1	85.1	87.9	89.9	68.9	69.6	76.9	**84.3**	80.6	77.5

Table 4: Ablation studies on our pseudo box generation module and naive Siamese tracker initialization before cycle memory training on the VOT2017/18 dataset.

Operations	A ↑	R ↓	EAO ↑
Random boxes	0.488	0.646	0.195
Our gnereated boxes	**0.567**	**0.520**	**0.263**
w/o naive Siamese learning	0.575	0.389	0.306
w/ naive Siamese learning	**0.578**	**0.304**	**0.344**

Table 5: Quantitative results on the IoU success rates of the pseudo boxes in template frames and memory frames.

Dataset \ IoU	Template		Memory	
	0.3	0.5	0.3	0.5
GOT-10k	63.2%	45.5%	62.0%	43.8%
VID	64.4%	42.1%	63.9%	42.0%

Pseudo Bounding Box Generation. To better investigate the precision of the pseudo bounding boxes, we collect over 10^4 training instances and compute the IoU scores between the output pseudo bounding boxes and the ground truth bounding boxes on both the GOT-10k and VID datasets. Tab. 5 shows the success rates of the pseudo bounding boxes over different IoU scores in both template frames and memory frames. On both datasets, over 63% sampled boxes in template frames cover at least parts of the foreground objects ($IoU > 0.3$), while over 42% sampled boxes in template frames are precise enough to cover approximately the entire objects ($IoU > 0.5$). Besides, from the small difference between the IoU success rates of pseudo boxes in template frames and memory frames on both datasets, we conclude that using large frame intervals for cycle memory training only slightly decreases the reliability of memory frames compared to template frames. This explains why our unsupervised tracker can learn from large motions.

Training Dataset. Since most existing unsupervised deep trackers are trained on the VID dataset, we investigate the impact of training data on USOT* on the VOT2017/18 benchmark. As is shown in Tab. 6, when only using VID as the training set, the proposed tracker still achieves 0.315 in EAO, with an 8.5 points increase over the state-of-the-art unsupervised tracker LUDT+ (i.e., 0.230 in EAO). Besides, our tracker benefits from training on more unlabeled videos, inferring the great potential of unsupervised tracking.

Table 6: Ablation studies on training data. With more unlabeled videos used for training, the proposed USOT* achieves better results on the VOT2017/18 dataset.

Training Data				A ↑	R ↓	EAO ↑
VID	GOT-10k	LaSOT	YT-VOS			
✔				0.576	0.337	0.315
✔	✔			**0.587**	0.323	0.320
✔	✔	✔		0.579	0.328	0.337
✔	✔	✔	✔	0.578	**0.304**	**0.344**

Table 7: Parameter sensitivity of the length and weight of the online memory queue on the VOT2017/18 dataset.

$N_q \setminus w$	0.3	0.5	0.6	0.7	0.8
5	0.289	0.302	0.323	0.313	0.323
6	0.294	0.312	0.312	0.329	0.322
7	0.310	0.318	0.336	**0.344**	0.331
8	0.302	0.300	0.319	0.341	0.338

Online Update. We study the parameter sensitivity of N_q and w in the online memory module. N_q indicates the number of memorized features collected online in the memory queue, while w indicates the weight for $\mathcal{R}_{mem}$. Tab. 7 reports the EAO scores of USOT* on the VOT2017/18 dataset. The cooperation of offline and online modules with $w = 0.7$ benefits the proposed tracker most, and setting the length of the memory queue N_q to 7 is most suitable.

5. Concluding Remarks

In this paper, we propose learning a robust tracker from unlabeled videos from scratch. We first generate candidate box sequences to cover moving objects in videos. We then train a naive Siamese tracker using single-frame pairs. We finally continue training the naive tracker in longer temporal spans with a novel cycle memory scheme, enabling the tracker to update online. Extensive experiments demonstrate that the proposed unsupervised tracker sets new state-of-the-art unsupervised tracking results, and even performs on par with recent supervised deep trackers. This work unveils the power of unsupervised learning for object tracking.

Acknowledgements. This work was supported by NSFC (61906119, U19B2035), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and Shanghai Pujiang Program.

References

[1] Boris Babenko, Ming-Hsuan Yang, and Serge J. Belongie. Robust object tracking with online multiple instance learning. *TPAMI*, 2011.

[2] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshop*, 2016.

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

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, 2020.

[5] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*, 2020.

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

[7] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, 2017.

[8] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: accurate tracking by overlap maximization. In *CVPR*, 2019.

[9] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[10] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Ann. Oper. Res.*, 2005.

[11] 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.

[12] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019.

[13] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, 2021.

[14] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[15] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*, 2020.

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

[17] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 2015.

[18] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 2019.

[19] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018.

[20] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman P. Pflugfelder, Luka Cehovin, Tomás Vojír, Gustav Häger, Alan Lukezic, Gustavo Fernández, et al. The visual object tracking VOT2016 challenge results. In *ECCV Workshop*, 2016.

[21] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman P. Pflugfelder, Joni-Kristian Kämäräinen, Martin Danelljan, Luka Cehovin Zajc, Alan Lukezic, Ondrej Drbohlav, Linbo He, et al. The eighth visual object tracking VOT2020 challenge results. In *ECCV Workshop*, 2020.

[22] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman P. Pflugfelder, Luka Cehovin Zajc, Tomás Vojír, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, Gustavo Fernández, Álvaro García-Martín, Álvaro Iglesias-Arias, et al. The sixth visual object tracking VOT2018 challenge results. In *ECCV Workshop*, 2018.

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

[24] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1989.

[25] 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.

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

[27] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*, 2019.

[28] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, 2020.

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

[30] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[31] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[32] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *IJCV*, 2008.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[34] Chon-Hou Sio, Yu-Jen Ma, Hong-Han Shuai, Jun-Cheng Chen, and Wen-Huang Cheng. S2siamfc: Self-supervised fully convolutional siamese network for visual tracking. In *ACM MM*, 2020.

[35] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson W. H. Lau, and Ming-Hsuan Yang. VITAL: visual tracking via adversarial learning. In *CVPR*, 2018.

[36] Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: visual tracking by re-detection. In *CVPR*, 2020.

[37] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, 2019.

[38] Ning Wang, Wengang Zhou, Yibing Song, Chao Ma, Wei Liu, and Houqiang Li. Unsupervised deep representation learning for real-time tracking. *IJCV*, 2021.

[39] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv:1704.04057*, 2017.

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

[41] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[42] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 2015.

[43] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian L. Price, Scott Cohen, and Thomas S. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

[44] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas S. Huang. Unitbox: An advanced object detection network. In *ACM MM*, 2016.

[45] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R. Scott. Deformable siamese attention networks for visual object tracking. In *CVPR*, 2020.

[46] 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.

[47] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 2019.

[48] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020.

[49] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *AAAI*, 2020.

[50] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018.