

# Unsupervised Learning of Accurate Siamese Tracking

Qihong Shen<sup>1,†</sup>, Lei Qiao<sup>2</sup>, Jinyang Guo<sup>3</sup>, Peixia Li<sup>3</sup>, Xin Li<sup>4</sup>,  
Bo Li<sup>2</sup>, Weitao Feng<sup>3</sup>, Weihao Gan<sup>2,5</sup>, Wei Wu<sup>2,5</sup>, Wanli Ouyang<sup>3,5</sup>

<sup>1</sup> Harbin Institute of Technology (Shenzhen) <sup>2</sup> SenseTime Research

<sup>3</sup> The University of Sydney <sup>4</sup> Peng Cheng Laboratory <sup>5</sup> Shanghai AI Laboratory

{shenqihong0905,xinlihitzsz}@gmail.com, {qiaolei,libo,ganweihao,wuwei}@sensetime.com,  
{jinyang.guo,peixia.li,weitao.feng,wanli.ouyang}@sydney.edu.au

## Abstract

*Unsupervised learning has been popular in various computer vision tasks, including visual object tracking. However, prior unsupervised tracking approaches rely heavily on spatial supervision from template-search pairs and are still unable to track objects with strong variation over a long time span. As unlimited self-supervision signals can be obtained by tracking a video along a cycle in time, we investigate evolving a Siamese tracker by tracking videos forward-backward. We present a novel unsupervised tracking framework, in which we can learn temporal correspondence both on the classification branch and regression branch. Specifically, to propagate reliable template feature in the forward propagation process so that the tracker can be trained in the cycle, we first propose a consistency propagation transformation. We then identify an ill-posed penalty problem in conventional cycle training in backward propagation process. Thus, a differentiable region mask is proposed to select features as well as to implicitly penalize tracking errors on intermediate frames. Moreover, since noisy labels may degrade training, we propose a mask-guided loss reweighting strategy to assign dynamic weights based on the quality of pseudo labels. In extensive experiments, our tracker outperforms preceding unsupervised methods by a substantial margin, performing on par with supervised methods on large-scale datasets such as TrackingNet and LaSOT. Code is available at <https://github.com/FlorinShum/ULAST>.*

## 1. Introduction

Visual tracking has become an integral part of various video applications such as autonomous driving and video recognition. In the mainstream of visual object tracking, deep learning-based trackers are dominant [28], requiring

a large number of labeled videos. Since the labeled data occupy a relatively small portion of practical scenes, the trained tracker cannot reliably track previously unseen objects. Thus, learning from unlabeled videos becomes a promising approach. Prior works on unsupervised tracking fall into two categories: exploiting self-supervision signals in videos from either the spatial or temporal dimensions. For the first category [24, 33], the focus is on how to construct a template-search pair using a still frame. Since these methods are limited by the inability to learn temporal correspondence over long periods of time, trained trackers can no longer track objects with strong variation. To cope with the appearance variations that occur during online tracking, we center our attention on methods in the latter category, exploiting the temporal self-supervision signal in videos.

In supervised tracking methods, box-regression branch has been demonstrated to be effective to capture the objects with large scale variation along temporal dimension [4, 22]. However, in existing unsupervised methods, this branch is always absent [33, 34, 38]. Recently, USOT [47] introduced a box-regression head, but the tracker is initially trained with template-search pairs from single frames, followed by cycle memory training to enhance the robustness of classification branch, whereas the box-regression branch is trained with spatial supervision alone. In this paper, we aim to train a better tracker by learning temporal correspondence both on the classification branch and on the regression branch. However, we identify there are three critical challenges.

First, despite unlimited self-supervision can be obtained by tracking a video along a cycle in time, how to explore the self-supervision signal in temporal dimension of videos for training a tracker equipped with a box-estimation branch is not well explored in existing methods. In cycle training, as illustrated in Fig. 1, a tracker is assumed to be capable of tracking back to the initial location. The tracker is evolved in the cycle by utilizing the inconsistency between start and end location in initial frames. However, when training from

<sup>†</sup>Work performed when Qihong Shen is an intern at SenseTime.

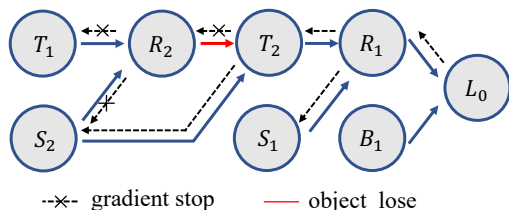


Figure 1. Three challenges when learning a better tracker with temporal correspondence on both classification and regression branches. First, in forward propagation process, tracker often loses target objects in intermediate frames, breaking down the training pipeline. Second, in backward propagation process, gradient cannot flow through the whole framework due to the RoI-Align. Third, the pseudo labels are often noisy, degrading the performance. In this figure,  $T$ ,  $S$ ,  $R$ ,  $B$ , and  $L$  denote template kernel, search region, candidate boxes, pseudo labels, and loss value, respectively. The subscript indexes the temporal order of frames.

scratch, it is hard for the tracker to find the target object. The template kernel generated on the intermediate frames is likely to not contain any features of the target object, which means the tracker cannot return to the initial location. The training pipeline will break down after several iterations.

Second, we identify misalignment in cycle training: when the tracking result  $R_2$  is inaccurate, the generated template kernel  $T_2$  is likely to embrace many distractor features, but the imposed loss still forces the tracker to predict accurate target box by using such noisy template, which causes an ill-posed penalty. From the gradient flow of cycle training, we can observe the select operation like RoI-Align, since the coordinate is quantized, is not differentiable on boxes coordinate. Therefore the gradient cannot back propagate to the node before  $R_2$ . In other words, the tracking errors on intermediate frames cannot be penalized in this pipeline.

Third, as unsupervised tracking framework still relies on the pseudo labels in initial frames and box-regression branch training requires objects that have clear edges, pseudo labels in initial frames are crucial. However, we observe these labels are likely to be noisy, degrading the tracking performance.

To address the aforementioned three challenges, we propose a novel unsupervised tracking framework called ULAST, which aims to learn temporal correspondence both on the classification branch and regression branch. It consists of three newly proposed components: consistency propagation transformation, region mask operation, and mask-guided loss re-weighting. Specifically, the consistency propagation transformation aims to generate reliable template kernel for tracking next frame, which uses both long-term and short-term information from template kernels and search regions of previous frames. As a result, it enables our framework to exploit temporal self-supervision signal and avoiding the training pipeline breaking down. Instead of using RoI-Align, our region mask operation selects

features with all candidates in  $R_2$  based on the search region feature and predicted bounding boxes from previous frame, and makes regression and classification heads differentiable to implicitly penalize tracking errors on intermediate frames. The mask-guided loss re-weighting strategy dynamically assigns weights to samples based on the quality of their pseudo labels, which avoid using the noisy pseudo labels.

We evaluate the trained tracker on five diverse benchmark datasets, and the favourable performance against state-of-the-art methods demonstrate the effectiveness of our proposed framework. The main contribution of this work are summarized as follows:

- We propose a novel unsupervised learning framework called ULAST, which can learn temporal correspondence both on classification and regression branches.
- A consistency propagation transformation is proposed to generate reliable template kernel, avoiding the training process of our ULAST framework breaking down.
- A differentiable region mask operation is proposed to select features as well as implicitly penalize the tracking errors of intermediate frames in backward propagation process.
- A mask-guided loss re-weighting strategy is proposed to mitigate the negative impact of noise on training.

## 2. Related work

**Supervised visual tracking** The past few years have witnessed significant performance improvement in deep-learning based trackers. We can roughly divide these trackers into two categories: online-optimized trackers [2, 4–6] and offline trackers [3, 9, 10, 21, 41, 43, 46]. Online-optimized trackers rely on online update methods with dedicated design. The rough position of target objects are figured out by ridge regression with updated template kernels. Then refinement is applied to estimate accurate bounding boxes. On the other hand, offline trackers learn to match the template and search region in metric space instead. This category is dominated by Siamese-network based trackers. The pioneering work SiamFC [1] extracts features of template and search patches with shared backbone network. Then cross correlation is applied to generate response map for locating the target. Various efforts have been made in this category, such as better backbone networks [21], target-aware attentions [23, 43], anchor-free regression [10, 46], and effective template-search fusion [13, 45]. Efficient tracking [29, 42] with Siamese-based trackers is also explored with pruning [11, 12] and network architecture search [31]. Nevertheless, all of these methods require extensive supervised training involving a large number of annotated videos to learn the correspondence between template and search region. In contrast, our work is an unsupervised learn-

ing framework for accurate Siamese-based tracking, which does not require a huge annotation cost.

**Unsupervised visual tracking.** Since it is costly to collect annotations for videos, unsupervised tracking [33,35,38,47] becomes a promising approach for training more robust trackers. The pioneering work UDT [34] trained a Discriminative Correlation Filters (DCF) based tracker by forward-backward tracking frames with the supervision of consistency loss. These works indicated that cycle consistency in videos can be utilized to effectively train a robust tracker by forward-backward tracking multiple frames. On the other hand,  $s^2$ siamfc [33] proposed a Siamese network based unsupervised training framework by mining the self-supervision in single frames, in which adversarial masking is learnt to construct template-search pairs from identity frame. However, tracking performance of these methods [34, 47] depends heavily on online updating schemes. Without online updates, these unsupervised trained trackers cannot handle challenging objects with dramatic variation. Recently, acquiring self-supervision signals from both spatial and temporal dimension becomes the promising way for unsupervised tracking. Zheng *et al.* [47] proposed an unsupervised training approach by naive training from single frame in first stage, then cycle training is adopted to learn on longer temporal spans. PUL [38] proposed to initially learn a background discrimination model by contrastive learning. Then the model continues training with temporal corresponding patches mined with a noise-robust loss. Besides, various pretext tasks [17,27,36] are constructed in learning visual representation from videos by utilizing the forward-backward tracking idea. Different from these works, our framework mainly focuses on learning classification and regression capability simultaneously from the temporal supervision, achieving superior tracking performance.

### 3. Proposed Method

#### 3.1. Preliminary

**Siamese-network based Tracker.** To handle scale variation of the target in the cycle, our ULAST is built upon Siamese-network based region proposal network [21]. Suppose we need to use a template patch to find the target object in a search region. The tracker first use a ResNet50 with shared parameter to extract the feature from the template and search region, and we denote the extracted feature from the template and search region patches as  $T$  and  $S$ , respectively. Then, a region proposal network is learnt to generate the bounding box and the corresponding class. The network is optimized with classification loss and regression loss:

$$L^l = \lambda_1 L_{cls}^l + \lambda_2 L_{reg}^l, \quad (1)$$

where  $L_{cls}^l$  and  $L_{reg}^l$  are respectively the Focal loss [25] and the  $L_1$  loss.  $\lambda_1$  and  $\lambda_2$  are coefficient to balance two terms,

the superscript of  $L^l$  denotes the conventional (legacy) loss in Siamese pair training paradigm [22].

**Cycle Training.** Suppose we track three sampled frames in a cycle with the given pseudo-label of the first frame. when tracking the 2nd frame, we use the template kernel extracted from patch in the 1st frame to predict candidate boxes on search region. Then these boxes are leveraged to generate new template kernel by selecting feature from search region. When tracking subsequent frames, we use template kernels generated from last search region to track. Generally, we track frames in palindrome order (i.e frames are ordered as 1,2,3,2,1) to track the objects back to the first frame. The tracker can be optimized by inconsistency the final tracking result and the pseudo label in the first frame. Formally, we call this training pipeline cycle training, as opposed to legacy training from single frames. The formulation of loss  $L^c$  in cycle training is same as that in legacy training, which consists of Focal loss [25] and  $L_1$  loss. Thus, the total loss in cycle training is formulated as:

$$L_{total} = (1 - \lambda_c)L^l + \lambda_c L^c, \quad (2)$$

Here  $L^l$  is the loss for self-tracking, constructing template-search pairs from first frames. The cycle loss  $L^c$  is computed by tracking back to the first frame in the cycle,  $\lambda_c$  is a manual weight parameter.

#### 3.2. Overview

The architecture of our framework is shown in Fig. 2, in which we use with three sparse sampled video frames for illustration. Given a palindromic video sequence, we first use a shared ResNet50 as the backbone to extract features from template of the first frame and search patches from the second frame and generate the template feature  $T_1$  and search region feature  $S_2$ . Then, the extracted feature  $T_1$  and  $S_2$  are fed into the regression branch and classification branch of the region proposal network (RPN) to yields box-regression result  $P_{reg}$  and corresponding classification confidence score  $P_{cls}$ . Then, we pass  $P_{cls}$  and  $P_{reg}$  to the region mask operation and generate the regional mask  $M_2$ . After that, our CPT module takes the search feature  $S_2$  and corresponding regional mask  $M_2$  as input to generate the template feature for the second frame  $T_2$ . At the same time, search region feature  $S_3$  is generated from 3rd frame. This search feature and the template feature  $T_2$  are then fed into RPN to predict box classification and regression results on 3rd frame. This process is repeated and finally generate the tracking result at the first frame, which will be then used to calculate the mask-guided loss based on the pseudo-label at the first frame.

#### 3.3. Training with cycle-consistency

**Region mask.** As introduced before, conventional cycle training suffered from the ill-posed penalty. As the top-

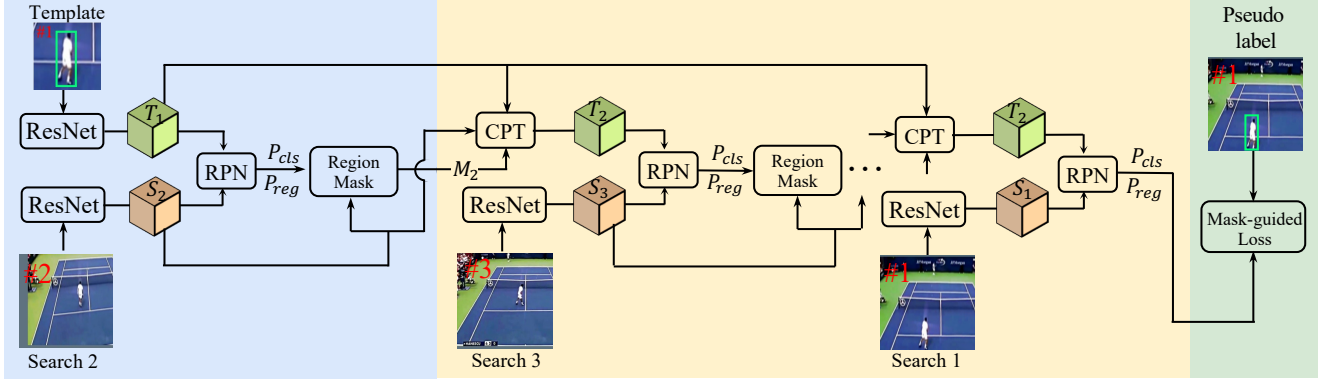


Figure 2. **Overview of our Framework.** Here we illustrates the overall framework with tracking 3 frames in  $1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 1$  order.

n prediction results of classification and regression branch are likely to be inaccurate in initial training stage. Generated new templates feature based on these top-n boxes may do not contain any target object feature. Forcing such template to track back to the initial target position leads to the ill-posed penalty. And the feature selection operation like RoI-Align is not differentiable on boxes coordinates, causing intermediate frames tracking error cannot be penalized. Although Precise RoI-Pooling [18] can be used to make the coordinate differentiable, it cannot pass the gradient to all the candidate boxes. To address these issues, it is necessary to introduce a module to that can select target features from last search region involving all estimated boxes and be differentiable on the coordinates.

Therefore, we propose a novel operation called region mask to select region-wise features and make the classification and regression branches differentiable in cycle training. Suppose we need to calculate the region mask based on the search region feature  $S_t \in \mathbb{R}^{C \times H \times W}$ , here subscript  $t$  denotes  $t$  th frame in sampled frames. The input of the region mask operation is the output of the RPN, which consists of the output  $P_{cls}$  and  $P_{reg}$  from the classification and regression branch, respectively. We first introduce a grid of size  $H \times W$ , same as the spatial resolution of search region feature  $S_t$ . In Fig 3, we illustrate the region mask operation when using a grid of  $4 \times 4$  size. Let us suppose there are totally  $K$  estimated box in the output of the regression branch

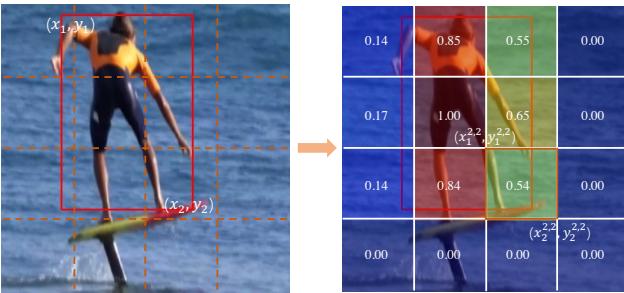


Figure 3. **Region mask operation.** We illustrate a grid map computed for one predicted box on search region. For better presentation, the grid size is set as  $4 \times 4$ . Better viewed with zoom in.

$P_{reg}$  and the grid map for the  $k$  th predicted box is denoted as  $G_k$ . Denotes the  $k$  th predicted box as  $(x_1, y_1, x_2, y_2)$  and the grid box at the position  $(i, j)$  as  $(x_1^{ij}, y_1^{ij}, x_2^{ij}, y_2^{ij})$ . The grid value  $G_k^{(i,j)}$  for the grid map  $G_k$  at  $i$  th row and  $j$  th column can be calculated as follows:

$$G_k^{(i,j)} = \frac{(\hat{x}_2^{ij} - \hat{x}_1^{ij})(\hat{y}_2^{ij} - \hat{y}_1^{ij})}{(x_2^{ij} - x_1^{ij})(y_2^{ij} - y_1^{ij})}, \quad (3)$$

where  $(\hat{x}_1^{ij}, \hat{y}_1^{ij})$  and  $(\hat{x}_2^{ij}, \hat{y}_2^{ij})$  are the top-left and bottom-right intersection corner,  $\hat{x}_1^{ij} = \max(x_1, x_1^{ij})$ ,  $\hat{y}_1^{ij} = \max(y_1, y_1^{ij})$ ,  $\hat{x}_2^{ij} = \min(x_2, x_2^{ij})$  and  $\hat{y}_2^{ij} = \min(y_2, y_2^{ij})$ . Here, the grid value represents the overlap ratio of the fixed grid with predicted box. Obviously, this formulation is naturally differentiable on the coordinates like IoU (Intersection of Union). Therefore, the grid map for all the boxes in this search region can be collected into a set  $\{G_k\}$ , for  $k = 1, 2, \dots, K$ .

As each bounding box predicted by the RPN has a corresponding confidence score, it is intuitive to combine these grid maps with their confidence. After getting the grid map set  $\{G_k\}$ , we aggregate it into a single channel regional mask. For the same spatial grid with duplicated grid values, we only use the grid value from the  $k$  th predicted box with the highest confidence score, and set the grid value from the other predicted box as 0. With this processing, we generate a new grid map  $\{\tilde{G}_k\}$ , at most only one grid map  $\tilde{G}_k$  has grid value greater than 0 across  $K$  grid maps for a certain spatial pixel  $(i, j)$ . For final aggregation, the regional mask of the search region feature  $S_t$  can be calculated as follows:

$$M_t = \sum_{k=1}^K \mathbb{1}(s_k, TH) \cdot s_k \tilde{G}_k, \quad (4)$$

where the indicator function is defined as:

$$\mathbb{1}(s_k, TH) \begin{cases} 1, & \text{if } s_k \geq TH \\ 0, & \text{otherwise} \end{cases}. \quad (5)$$

Here,  $M_t \in \mathbb{R}^{H \times W}$  is the generated regional mask.  $TH$

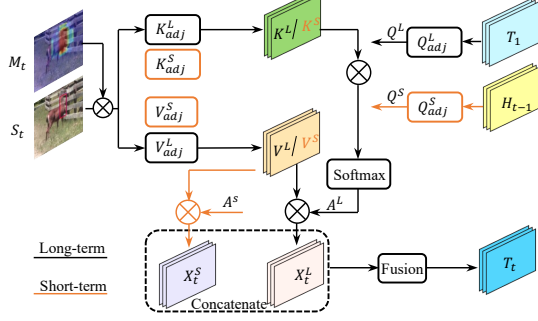


Figure 4. **Consistency propagation transformation.** In this transformation, we generate reliable template kernel from the last search region by retrieving long-shot term features. Black lines denote long-term target feature retrieval while orange lines denote short-term target feature retrieval.

is a manually set threshold.  $s_k$  denotes the classification confidence score for  $k$ th grid map.

**Consistency propagation transformation (CPT).** As the aforementioned pipeline broken problem, it is crucial to ensure consistency propagation between frames, i.e., the tracker should not lose the target object. In our framework, we argue that the reason for this problem is the lack of exploration of temporal consistency when introducing the box-estimation branch. The predicted boxes of subsequent frames will be largely affected by those from current frames. Namely, once the predicted box in the current frame is inaccurate, it is difficult to generate accurate predicted boxes in the subsequent frames. Therefore, we propose a consistency propagation transformation (CPT) module to exploit the self-supervision signal in the temporal dimension. Specifically, as illustrated in Fig. 4, our CPT module aims to retrieve a new template kernel  $T_t$  from search region feature  $S_t$  with regional mask  $M_t$  based on the predicted result on  $t$ th search frame. Specifically, to squeeze the noisy features in the search region feature denoted as  $S_t$ , a long-short term consistency module is introduced to retrieve features from that. As the initial template feature denoted as  $T_1$  comprises the most reliable features of the target, we get the long-term feature conditioned on  $T_1 \in \mathbb{R}^{C \times h \times w}$ . First, the region mask  $M_t \in \mathbb{R}^{H \times W}$  is multiplied with the search feature  $S_t \in \mathbb{R}^{C \times H \times W}$  to generate a pre-selected search feature denoted as  $\tilde{S}_t$ :

$$\tilde{S}_t = S_t \otimes M_t, \quad (6)$$

where  $\otimes$  denotes the element-wise multiplication. Then an  $1 \times 1$  convolution denoted as  $K_{adj}^L$  is applied to  $\tilde{S}_t$  for generating the long-term key features  $K^L \in \mathbb{R}^{C \times H \times W}$ . We also use another convolution denoted as  $Q_{adj}^L$  to apply on the template feature  $T_1$  and generate the long-term query feature  $Q^L \in \mathbb{R}^{C \times h \times w}$ . The superscript  $L$  denotes the long-term. Then, we reshape the query and key into the shape of  $C \times N_z$  and  $C \times N_x$  respectively, where  $N_z = h \times w$

and  $N_x = H \times W$ . Therefore, We can generate a long-term affinity matrix  $A^L \in \mathbb{R}^{N_z \times N_x}$  as:

$$A^L = \text{Softmax}_{col}((Q^L)^T K^L) \in \mathbb{R}^{N_z \times N_x}, \quad (7)$$

where  $\text{Softmax}_{col}$  is the softmax operation along the column dimension. We also use another  $1 \times 1$  convolution denoted as  $V_{adj}^L$  to adjust  $\tilde{S}_t$  for generating the long-term value features  $V^L \in \mathbb{R}^{C \times N_x}$ , which will be multiplied with the affinity matrix to generate the long-term feature  $X_t^L$ :

$$X_t^L = \text{reshape}(A^L (V^L)^T) \in \mathbb{R}^{C \times h \times w}. \quad (8)$$

For the short-term feature generation, the only difference from that for long-term is the input of the query adjust operation is different. We use a hidden template  $H_{t-1} \in \mathbb{R}^{C \times h \times w}$  as the input of the query adjust operation and generate the short-term query feature, where  $H_{t-1}$  is calculated when tracking the last frame:

$$H_{t-1} = f_\phi(\text{concat}(X_{t-1}^L, T_1)), \quad (9)$$

where  $\text{concat}(\cdot)$  is the concatenation operation.  $f_\phi$  denotes the hidden template aggregation module, which is a conv-bn block with learnable parameter  $\phi$ .

Except for the difference mentioned above, the other operations are the same as those for long-term feature generation. We use the similar manner to generate the short-term feature  $X_t^S$ .

Finally, with the long-term and short-term feature retrieved from current search feature  $S_t$ , we generate the output of the CPT module, which can be formulated as:

$$T_t = h_\theta(\text{concat}(X_t^S, X_t^L)). \quad (10)$$

Here,  $h_\theta$  denotes the aggregation operation implemented by using a conv-bn block with the learnable parameter  $\theta$ .

### 3.4. Learn from noise label

In our ULAST, we use the unsupervised trained optical flow model [26] to generate the pseudo labels of the first frame, which will be used in the training process. However, these pseudo labels often are noisy, which hinders the performance of the trained tracker. Existing approaches [33,38] only use the classification confidence from the response map to filter out label noise. We argue that the classification result is not sufficient to evaluate the importance of each sample. Therefore, we also take the regression result into consideration. Intuitively, template kernels generated from pseudo labels with more noise often produce larger response area from background in search regions, which can be identified from Fig 5. To this end, we propose a mask-guided loss re-weighting strategy to re-weight the loss from the pseudo-labels of various qualities. Specifically, suppose we have a video batch of size  $B$ . For the

$b$ -th sample in this batch, we first use the tracker to generate the classification output  $P_{cls}$  and regression output  $P_{reg}$  by constructing a template-search pair from the first frame. Then, based on  $P_{cls}$  and  $P_{reg}$ , we get the regional mask  $\hat{M}^b$  by using Eq. 3, 4 and 5 for this video. On the other hand, given the pseudo-labelled first frame, we also calculate  $\bar{M}^b$ , the only difference is that we set  $K = 1$  and  $s_k = 1$  in Eq. 4, as there is only one ground-truth box in the first frame in this case. Then we learn a dynamic weight  $w_b$  for each sample in a batch according to the relatively high response region in such formulation:

$$w_b = \log_\gamma \left( \alpha - \frac{\sum_p \sum_q \mathbb{1}(\hat{M}_{p,q}^b, \beta)}{\sum_p \sum_q \mathbb{1}(\bar{M}_{p,q}^b, \beta)} \right), \quad (11)$$

where  $\mathbb{1}$  is an indicator function defined as:

$$\mathbb{1}(\hat{M}_{p,q}^b, \beta) \begin{cases} 1, & \text{if } \hat{M}_{p,q}^b \geq \beta \\ 0, & \text{otherwise} \end{cases}. \quad (12)$$

Here,  $\hat{M}_{p,q}^b$  and  $\bar{M}_{p,q}^b$  are the grid at the  $p$ -th row and  $q$ -th column of the regional mask for the  $b$ -th sample in this batch.  $\gamma$  denotes the factor to scale the value of  $w_b$ ,  $\alpha$  denotes threshold for assigning weights to noisy labels, and  $\beta$  denotes threshold to filter out distractors of low response.

Therefore, the loss function can be formulated as:

$$L = \frac{1}{B} \sum_b w_b (\lambda_1 L_{cls} + \lambda_2 L_{reg}), \quad (13)$$

where  $L_{cls}$  and  $L_{reg}$  are the classification loss and regression loss introduced in Eq. 1. Here we omit superscript of  $L$  as this formulation is both applied on  $L^l$  and  $L^c$ .

### 3.5. Online tracking

As legacy training and cycle training are both adopted in offline training phase, the input of template kernel in RPN

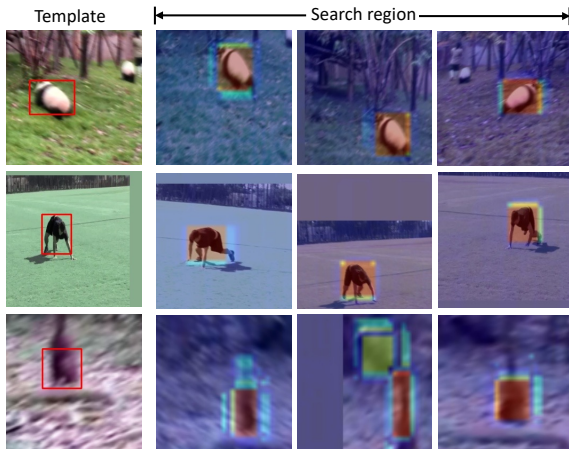


Figure 5. **Visualization of regional mask.** The pseudo label sample in the 3rd row encompasses only part of the target object, which is noisy for training the tracker, resulting in regional mask with larger high response area (green part in the 3rd row frames).

is compatible with template feature extracted from patches or feature retrieved from historical search feature by CPT module. For real-time tracking, our tracker can perform like SiamRPN [21], operating at a high speed about 80 FPS. For robust tracking, cycle training enables the tracker to update with a memory queue. Detailly, a memory queue of length  $N_L$  stored with historical search features and region masks is maintained, including features of initial frame and  $N_L - 1$  historical samples of the highest score. Once the memory queue is updated, the memory kernel will be updated. Besides, the hidden template  $H_t$  is updated every  $N_s$  frames with the highest score in this short interval. For the trade-off between speed and accuracy, the classification map  $R_{cls}$  from the legacy kernel and memory kernel is combined with  $R_{cls} = (1 - \lambda_m)R_{cls}^L + \lambda_m R_{cls}^M$ ,  $\lambda_m$  denotes the weight to balance the classification score, while the regression map is generated with the template kernel extracted from the initial template patch. Here,  $R_{cls}^L$  denotes the response map in classification branch that applying the template kernel extracted from initial frame, while  $R_{cls}^M$  is the classification response map when applying the template kernel extracted from online memory queue. Specifically, We set  $N_L = 6$  and  $N_s = 10$  throughout all online updating experiments.

## 4. Experiments and Results

### 4.1. Implementation details

**Data preparation.** The labels of our training are generated by using the off-the-shelf unsupervised optical flow model [26] on datasets Got10k [16], LaSOT [8], VID [32] and YoutubeVOS [40], the data sampling strategy is similar to USOT [47]. One reliable template frame and three search region frames with large temporal gaps are sampled from a video for cycle training. The template patch is cropped as size  $127 \times 127$ , and the search frames are replicated as palindromes for tracking back to the initial frame. The spatial size of all input frames is resized as  $640 \times 480$  without crop. Specifically, our framework only requires objects' initial box and subsequent center position, reducing the reliance on pseudo-labels. As the optical flow method struggles with scale variation of target objects, in the cycle training, the search frames are cropped as  $255 \times 255$  according to the input object center and last frame estimated object scale instead of the scale initialized in pseudo-labels.

**Network architecture.** The architecture of our network follows conventional Siamese trackers. We adopt the ResNet50 as the feature extractor. Features from layer 2, 3, 4 are used as inputs of the region proposal network (RPN), and these features are interpolated to the same spatial resolution. Also, a learned weight is used to aggregate these three correlation maps. The spatial resolution of search feature size is  $31 \times 31$ , the grid size in region mask operation is the same as this. The anchor scale of RPN is set as 8, and the

Table 1. Evaluation results on TrackingNet, VOT2016 and VOT2018 benchmark datasets. Unsup denotes unsupervised training.

Tracker	Unsup	TrackingNet			VOT2016			VOT2018		
		Suc $\uparrow$	Pre $\uparrow$	NPre $\uparrow$	EAO $\uparrow$	Acc $\uparrow$	Rob $\downarrow$	EAO $\uparrow$	ACC $\uparrow$	Rob $\downarrow$
SiamFC [1]	No	0.571	0.533	0.663	0.235	0.532	0.461	0.188	0.503	0.585
DaSiamRPN [48]	No	-	-	-	0.411	0.610	0.220	0.326	0.560	0.340
SiamRPN++ [21]	No	0.733	0.694	0.800	-	-	-	0.414	0.600	0.234
ATOM [4]	No	0.703	0.648	0.711	-	-	-	0.401	0.590	0.204
DiMP [2]	No	0.740	0.687	0.801	-	-	-	0.440	0.597	0.153
KCF [15]	Yes	0.447	0.419	0.546	0.192	0.489	0.569	0.135	0.447	0.773
ECO [5]	Yes	0.561	0.489	0.621	0.375	0.550	0.569	0.280	0.270	0.480
S2SiamFC [33]	Yes	-	-	-	0.215	0.493	0.639	0.180	0.463	0.782
LU DT+ [35]	Yes	0.563	0.495	0.633	0.299	0.570	0.331	0.230	0.490	0.412
USOT [47]	Yes	0.599	0.551	0.682	0.351	0.593	0.336	0.290	0.564	0.435
USOT* [47]	Yes	0.616	0.566	0.691	0.402	0.600	0.233	0.344	0.578	0.304
ULAST*-off	Yes	0.649	0.585	0.725	0.397	0.599	0.224	0.347	0.569	0.304
ULAST*-on	Yes	0.654	0.592	0.732	0.417	0.603	0.214	0.355	0.571	0.286

anchor ratio is set as [0.33, 0.5, 1, 2, 3], ATSS [44] is used to assign anchor label with the top-15 candidates. There are  $K = 3125$  candidate boxes in the prediction result of RPN. **Training details.** Hyper-parameters for loss are set as  $\lambda_1 = 10$ ,  $\lambda_2 = 1.2$ ,  $\lambda_c = 0.5$ . In the region mask,  $TH$  is set as 0 to pass all predicted results. In the mask-guided re-weight strategy, we set hyper parameters as  $\gamma = 5$ ,  $\alpha = 7$  and  $\beta = 0.8s_{max}$ , where  $s_{max}$  denotes maximum score when generating the regional mask. The feature extractor of ResNet50 is initialized with Imagenet pretrain. The legacy training of 5 epochs is adopted to initialize the tracker with the ability to locate the rough position of the target object. Then 20 epochs of cycle training with one template frame and three search frames is adopted to enable the tracker to track objects with strong variation in practical cases. In the training process, we set the batch size as 8, and an SGD optimizer is used. The initial learning rate is set as  $1e^{-3}$ , which is gradually decayed to  $5e^{-5}$  in logspace.

## 4.2. Comparison with SOTA

We compare our proposed method with unsupervised and supervised methods on five challenging datasets, including OTB2015 [39], VOT2016 [19], VOT2018 [20], TrackingNet [30] and LaSOT [8]. The offline tracking mode is denoted as ULAST\*-off, while the online tracking mode with memory update is denoted as ULAST\*-on.

**VOT2016.** There are totally 60 videos in this dataset. In this benchmark, three metrics are used to report tracking performance: Robustness (Rob), Accuracy (Acc) and Expected Average Overlap (EAO) [19]. Table 1 shows evaluation results of our tracker. Without online update, the ULAST\*-off achieves better robustness score than the USOT\* with online update. When aided with online update, our ULAST\*-on achieves the best tracking performance among these unsupervised trackers according to these three metrics.

**VOT2018.** VOT2018 contains more challenging video sequences than VOT2016 datasets. As shown in Table 1, the ULAST\*-off has an EAO score of 0.347, which is superior

Table 2. Evaluation results on OTB2015 and LaSOT datasets.

Tracker	Unsup	OTB2015		LaSOT	
		Suc $\uparrow$	Pre $\uparrow$	Suc $\uparrow$	Pre $\uparrow$
SiamFC [1]	No	0.582	0.771	0.336	0.339
SiamRPN [22]	No	0.637	0.851	0.411	0.380
SiamRPN++ [21]	No	0.696	0.923	0.495	0.493
KCF [15]	Yes	0.485	0.696	0.178	0.166
DSST [7]	Yes	0.518	0.689	0.207	0.189
LU DT+ [35]	Yes	0.639	0.843	0.305	0.288
USOT [47]	Yes	0.589	0.806	0.337	0.323
USOT* [47]	Yes	0.574	0.775	0.358	0.340
ULAST*-off	Yes	0.645	0.878	0.468	0.448
ULAST*-on	Yes	0.648	0.879	0.471	0.451

to all unsupervised trackers without online update. Furthermore, ULAST\*-on can realize performance improvement with online update, achieving an EAO score of 0.355.

**OTB2015.** OTB2015 contains 100 video sequences with various targets. We compare our proposed method with eight representatives of supervised and unsupervised trackers, and evaluation results are shown in Table 2. ULAST\*-on achieves the best success and precision score among these unsupervised tracking with Success score of 0.648. And it is worth mentioning that our ULAST\*-off outperforms methods [35, 47] that rely on online update.

**TrackingNet.** TrackingNet is a large-scale benchmark for tracking in the wild. In addition to the precision and success metric used in OTB2015, TrackingNet introduced another metric called normalized precision (NPre). The evaluation results are shown in Table 1. ULAST\*-off achieves a Success score of 0.649, outperforming the state-of-the-art unsupervised methods by a large margin. And the ULAST\*-on achieves a gain of 0.5 on success score when applying the online update with the CPT module.

**LaSOT.** This benchmark dataset is the largest annotated one in the tracking community, consisting of 280 long videos sequences. The evaluation metric of this benchmark is the same as the TrackingNet, and the evaluation results are shown in Table 2. ULAST\*-off and ULAST\*-on

both achieve significant performance improvement on this benchmark with success score of 0.645 and 0.648.

### 4.3. Ablation Study and Algorithm Analysis

In this section, we conduct extensive experiments and give detailed analysis of our proposed method. For simplicity, region mask, region mask (detach head), Precise RoI-pooling, mask-guided loss re-weighting, residual connection are abbreviated as RM, RM(D), PrPool, ReLoss and Res respectively. Besides, models are all trained on LaSOT [8] and Got10k [16] with half iterations per epoch of the full version.

Table 3. Quantitative analysis of CPT module and region mask with ULAST\*-off on VOT2018 benchmark.

RM	RM(D)	PrPool	CPT	Res	Acc $\uparrow$	Rob $\downarrow$	EAO $\uparrow$
✓			✓		0.560	0.272	0.346
✓			✓	✓	0.560	0.304	0.317
	✓		✓		0.558	0.337	0.322
		✓	✓		0.552	0.342	0.301
		✓			-	-	-
			✓		0.561	0.309	0.310

**Consistency Propagation Transformation(CPT).** For evaluating the contribution of the CPT module, we conduct three experiments. Notably, the output of this module does not contain the common residual connection in attention-based operations. The motivation of this design is to exploit rich temporal feature of the target object, reducing the reliance on initial template feature. To validate that, we carry out an experiment trained with residual connection in CPT module by adding the initial template feature to the output. As the result shown in Table 3, the EAO score drops from 0.346 to 0.317 with this residual connection, demonstrating that involving the initial template will cause a performance drop. Besides, we remove the region mask to evaluate the CPT module alone, the EAO score drops to 0.310, but compared to tracker trained with single frame (in Table 4, EAO=0.296), i.e. spatial self-supervision only, this model still has a performance improvement of 1.4 EAO. In addition, we try to replace the CPT module with PrPool. As mentioned above, the training pipeline is prone to breaks down (gradient explosion) in this setting for the incorrect prediction result of the tracker in intermediate frames.

**Region mask.** To better understand the proposed region mask, here we perform two experiments. First, the region mask operation is substituted by the Precise RoI-pooling [18] to select features from last frame. We pool the candidate features from top-3 scored predicted boxes after non-maximum suppression. Pooled features are then fused by averaging, as shown in Table 3, the EAO score drops to 0.301. Second, we detach the gradient of candidate boxes on search region when generating the regional mask. In this form, the regional mask lost the ability to implicitly penalize the intermediate tracking errors. As the RM(D) option shown in Table 3, the EAO score drops from 0.346 to 0.322.

Table 4. Quantitative analysis of the mask-guided loss re-weighting on VOT2018 benchmark.

RM	CPT	ReLoss	Acc $\uparrow$	Rob $\downarrow$	EAO $\uparrow$
			0.559	0.370	0.284
		✓	0.565	0.361	0.296
✓	✓		0.557	0.309	0.331
✓	✓	✓	0.560	0.272	0.346

Table 5. Quantitative analysis of pretrain feature impact on OTB2015 and LaSOT benchmark. \* denotes the ULAST model trained with ImageNet pretrain.

setting	OTB2015		LaSOT	
	Suc $\uparrow$	Pre $\uparrow$	Suc $\uparrow$	Pre $\uparrow$
ULAST-on	0.610	0.811	0.433	0.407
ULAST-off	0.607	0.812	0.429	0.405
ULAST*-on	0.643	0.862	0.442	0.418
ULAST*-off	0.639	0.862	0.436	0.410

**Mask guided sample re-weighting.** Here we conduct experiments on cycle training and legacy training (training by template-search pairs from single frames) to validate the effectiveness of this strategy. As shown in Table 4, with this strategy, both of the performances are boosted in EAO: from 0.331 to 0.346 and from 0.284 to 0.296 for cycle training and legacy only training, respectively.

**Impact of ImageNet pretrain.** For fair comparison to existing unsupervised tracking works, here we train our tracker from scratch. Detailly, we substitute ResNet50 [14] pretrained on ImageNet classification as DenseCL [37] pretrain, which is self-supervised trained with contrastive learning. Then the trained model is evaluated on OTB2015 and LaSOT benchmark datasets for comparison, results are shown in Table 5. The performance of the ULAST-on and ULAST-off have a relatively slight performance drop compared to ULAST\* model. It suggests that a better representation can contribute to unsupervised visual tracking.

## 5. Conclusion

In this paper, we propose a novel framework for unsupervised tracking called ULAST. To generate reliable template kernel in the forward propagation process and thus enable our framework to be trained with cycle consistency, we first propose a consistency transformation. In the backward propagation process, we propose a region mask operation to implicitly penalize tracking error on intermediate frames. Besides, a mask-guided loss re-weighting strategy is proposed to assign dynamic weight to the loss from samples of various pseudo label qualities. With these proposed components, our ULAST can fully explore temporal supervision in unsupervised tracking process and achieves state-of-the-art performance.

**Acknowledgement:** Wanli Ouyang was supported by the Australian Research Council Grant DP200103223, Australian Medical Research Future Fund MRFAI000085, and CRC-P Smart Material Recovery Facility (SMRF) – Curby Soft Plastics.



## References

- [1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. [2](#), [7](#)
- [2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Int. Conf. Comput. Vis.*, 2019. [2](#), [7](#)
- [3] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021. [2](#)
- [4] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. [1](#), [2](#), [7](#)
- [5] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6638–6646, 2017. [2](#), [7](#)
- [6] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. [2](#)
- [7] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. Bmva Press, 2014. [7](#)
- [8] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. [6](#), [7](#), [8](#)
- [9] Dongyan Guo, Yanyan Shao, Ying Cui, Zhenhua Wang, Liyan Zhang, and Chunhua Shen. Graph attention tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2021. [2](#)
- [10] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6269–6277, 2020. [2](#)
- [11] Jinyang Guo, Wanli Ouyang, and Dong Xu. Channel pruning guided by classification loss and feature importance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10885–10892, 2020. [2](#)
- [12] Jinyang Guo, Wanli Ouyang, and Dong Xu. Multi-dimensional pruning: A unified framework for model compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1508–1517, 2020. [2](#)
- [13] Wencheng Han, Xingping Dong, Fahad Shahbaz Khan, Ling Shao, and Jianbing Shen. Learning to fuse asymmetric feature maps in siamese trackers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16570–16580, 2021. [2](#)
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. [8](#)
- [15] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015. [7](#)
- [16] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. [6](#), [8](#)
- [17] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020. [3](#)
- [18] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yunying Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision*, pages 784–799, 2018. [4](#), [8](#)
- [19] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomáš Vojír?, Gustav Häger, Alan Lukežič, Gustavo Fernández, et al. The visual object tracking vot2016 challenge results. 2016. [7](#)
- [20] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. 2018. [7](#)
- [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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. [2](#), [3](#), [6](#), [7](#)
- [22] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. [1](#), [3](#), [7](#)
- [23] Xin Li, Chao Ma, Baoyuan Wu, Zhenyu He, and Ming-Hsuan Yang. Target-aware deep tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. [2](#)
- [24] Xin Li, Wenjie Pei, Zikun Zhou, Zhenyu He, Huchuan Lu, and Ming-Hsuan Yang. Crop-transform-paste: Self-supervised learning for visual tracking, 2021. [1](#)
- [25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. [3](#)
- [26] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6489–6498, 2020. [5](#), [6](#)
- [27] Xingyu Liu, Joon-Young Lee, and Hailin Jin. Learning video representations from correspondence proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4281, 2019. [3](#)
- [28] Seyed Mojtaba Marvasti-Zadeh, Li Cheng, Hossein Ghanei-Yakhdan, and Shohreh Kasaei. Deep learning for visual

- tracking: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 1
- [29] Seyed Mojtaba Marvasti-Zadeh, Javad Khaghani, Li Cheng, Hossein Ghanei-Yakhdan, and Shohreh Kasaei. Chase: Robust visual tracking via cell-level differentiable neural architecture search. *arXiv preprint arXiv:2107.03463*, 2021. 2
- [30] 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 *Eur. Conf. Comput. Vis.*, 2018. 7
- [31] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pages 4095–4104. PMLR, 2018. 2
- [32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 6
- [33] 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 *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1948–1957, 2020. 1, 3, 5, 7
- [34] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1308–1317, 2019. 1, 3
- [35] Ning Wang, Wengang Zhou, Yibing Song, Chao Ma, Wei Liu, and Houqiang Li. Unsupervised deep representation learning for real-time tracking. *International Journal of Computer Vision*, 129(2):400–418, 2021. 3, 7
- [36] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019. 3
- [37] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. 8
- [38] Qiangqiang Wu, Jia Wan, and Antoni B Chan. Progressive unsupervised learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2993–3002, 2021. 1, 3, 5
- [39] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1834–1848, 2015. 7
- [40] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European conference on computer vision*, pages 585–601, 2018. 6
- [41] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. *arXiv preprint arXiv:2103.17154*, 2021. 2
- [42] Bin Yan, Houwen Peng, Kan Wu, Dong Wang, Jianlong Fu, and Huchuan Lu. Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15189, 2021. 2
- [43] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R. Scott. Deformable siamese attention networks for visual object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2
- [44] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9759–9768, 2020. 7
- [45] Zhipeng Zhang, Yihao Liu, Xiao Wang, Bing Li, and Weiming Hu. Learn to match: Automatic matching network design for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13339–13348, 2021. 2
- [46] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 2
- [47] Jilai Zheng, Chao Ma, Houwen Peng, and Xiaokang Yang. Learning to track objects from unlabeled videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13546–13555, 2021. 1, 3, 6, 7
- [48] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *Eur. Conf. Comput. Vis.*, 2018. 7