

## Article

# Generalized Hierarchical Co-Saliency Learning for Label-Efficient Tracking

Jie Zhao <sup>1</sup>, Ying Gao <sup>2,\*</sup>, Chunjuan Bo <sup>3</sup> and Dong Wang <sup>1</sup>

<sup>1</sup> Dalian University of Technology, Dalian 116024, China; zj982853200@dlut.edu.cn (J.Z.); wdice@dlut.edu.cn (D.W.)

<sup>2</sup> Henan Key Laboratory of Safety Technology for Water Conservancy Project, Henan Water & Power Engineering Consulting Co., Ltd., Zhengzhou 451450, China

<sup>3</sup> Dalian Minzu University, Dalian 116620, China; bcj@dlmu.edu.cn

\* Correspondence: gao6915319@163.com

## Abstract

Visual object tracking is one of the core techniques in human-centered artificial intelligence, which is very useful for human–machine interaction. State-of-the-art tracking methods have shown their robustness and accuracy on many challenges. However, a large amount of videos with precisely dense annotations are required for fully supervised training of their models. Considering that annotating videos frame-by-frame is a labor- and time-consuming workload, reducing the reliance on manual annotations during the tracking models’ training is an important problem to be resolved. To make a trade-off between the annotating costs and the tracking performance, we propose a weakly supervised tracking method based on co-saliency learning, which can be flexibly integrated into various tracking frameworks to reduce annotation costs and further enhance the target representation in current search images. Since our method enables the model to explore valuable visual information from unlabeled frames, and calculate co-salient attention maps based on multiple frames, our weakly supervised methods can obtain competitive performance compared to fully supervised baseline trackers, using only 3.33% of manual annotations. We integrate our method into two CNN-based trackers and a Transformer-based tracker; extensive experiments on four general tracking benchmarks demonstrate the effectiveness of our method. Furthermore, we also demonstrate the advantages of our method on egocentric tracking task; our weakly supervised method obtains 0.538 success on TREK-150, which is superior to prior state-of-the-art fully supervised tracker by 7.7%.

**Keywords:** visual tracking; weakly supervised learning; co-saliency attention; egocentric tracking



Academic Editors: Antonio Fernández-Caballero and Byung-Gyu Kim

Received: 20 June 2025

Revised: 23 July 2025

Accepted: 27 July 2025

Published: 29 July 2025

**Citation:** Zhao, J.; Gao, Y.; Bo, C.; Wang, D. Generalized Hierarchical Co-Saliency Learning for Label-Efficient Tracking. *Sensors* **2025**, *25*, 4691. <https://doi.org/10.3390/s25154691>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Visual object tracking (especially egocentric tracking) [1–3] is one of the core techniques in human-centered artificial intelligence [4]. Tracking any object of interest is very useful for human–machine interaction, which has many realistic applications (virtual reality, surveillance, vision control, to name a few). It has been confirmed that deeper neural networks are important to obtain better tracking performance, including convolutional neural networks (CNN) [5] and the recent Transformer networks [6]. Most of these deep network-based trackers are data-driven and fully supervised methods, since their training processes require plenty of training data with precise annotations. Some large-scale tracking datasets are developed to satisfy the training requirements for deep tracking models. For

instance, LaSOT [7] and GOT-10k [8] provide over 2.8 M and 1.4 M manual annotations for training, respectively. Each of their sequences is annotated and checked frame-by-frame by workers. However, such manually annotating videos frame-by-frame is quite time- and labor-consuming. It is a huge workload for larger video datasets collected on the Internet, where a steady stream of new videos will be uploaded every day. Therefore, label-efficient tracking is a problem worthy of studying, which focuses on effectively reducing requirements for precise annotations when training deep models, and enabling the model to make full use of unlabeled frames.

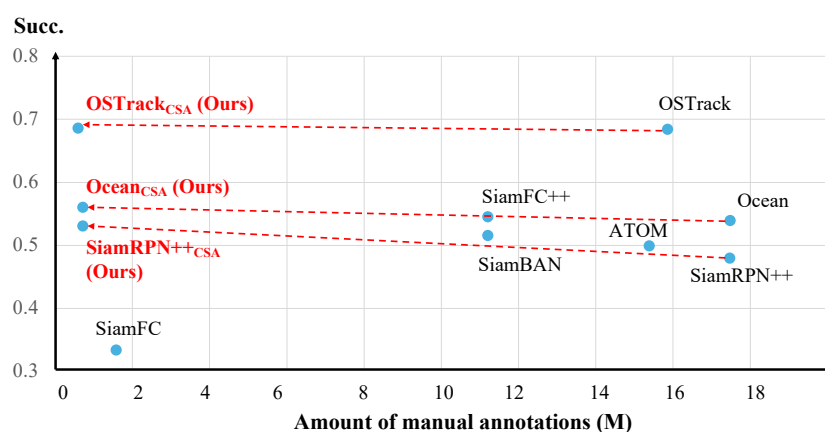
Despite being an important task, there are only few works exploring label-efficient tracking. Most of them attempt to investigate unsupervised tracking frameworks [9–11] without utilizing any annotation. Due to the lack of guidance for precise states of the specific target, these unsupervised trackers still have performance gaps with their baseline supervised methods. Further, Liu et al. [12] propose a reliable sample selection strategy for weakly supervised tracking, where poor-quality samples can be filtered by the proposed strategy. Approximately 9.13% manual annotations are required for training their models, and the obtained performance is close to 80% of full supervision. Different from these methods, in this paper, we propose a generalized co-saliency attention mechanism, which can be hierarchically integrated into various tracking frameworks to achieve weakly supervised tracking. Experiments demonstrate that our models require 3.33% annotations to achieve over 95% performance of full supervision.

The attention mechanism is a common and effective strategy for optimizing feature representations in many computer vision tasks. In [13], an attention-based selection mechanism is proposed to achieve feature filtering and fusion for salient detection. Liu et al. [14] propose a co-saliency attention mechanism to optimize the accuracy of person re-identification. Different from the motivations of these methods, we propose a co-saliency attention (CSA) module, which can mine rich co-salient visual information from unlabeled frames from both spatial and channel levels, thereby achieving weakly supervised tracking. The obtained co-salient attention map for the current search feature can enhance the common salient foreground information while weakening the background area. The proposed CSA can be treated as a generalized module that can be flexibly integrated into various tracking frameworks, achieving weakly supervised tracking and keeping the competitive tracking performance. Considering that the appearance and location of the target in a short clip have small changes, we adopt sparsely labeled datasets to train the model in a weakly supervised manner, that is, manually annotating one bounding box of the target every 30 frames (around one second for most videos). The remaining unlabeled frames will be arranged pseudo-labels by linear interpolation according to the first and last labels of the fragments. During training, 3.3% frames with precise labels will be sampled as template and search images, which are involved in the calculation of tracking losses, while those 96.7% unlabeled frames will be used as part of the input of our CSA module to assist in improving the current search features. This operation enables the model to make full use of the effective target information of unlabeled frames and save nearly 30 times the manual annotation costs.

As concluded in [15], human-centric consumer applications play a critical role in improving the quality of life. Due to the importance of these first-person vision tasks, egocentric tracking [1,2] has gradually obtained more and more attention in recent years. Compared with general VOT with third-person vision, challenges of egocentric tracking are reflected in rapid appearance changes, large camera motions, occlusions caused by frequent hand interactions, and so on. Most prior tracking methods have limited performance when facing this kind of sequences due to considering only the match between the initial frame and the current frame. In contrast, our CSA module enables the model to refer to rich visual

information from multiple frames, and enhance the target representation of the current frame by calculating co-saliency attention maps. Therefore, our weakly supervised method also provides a solution for egocentric tracking.

We hierarchically integrate our CSA into two CNN-based tracking models (anchor-based method SiamRPN++ [5] and anchor-free method Ocean [16]), and a Transformer-based tracking framework (OTrack [6]). Extensive experiments on four general VOT datasets and an egocentric tracking dataset can demonstrate the effectiveness and generalization of our weakly supervised tracking method on both third-person vision and first-person vision. An intuitive comparison between our weakly supervised methods and corresponding fully supervised baseline trackers is illustrated in Figure 1, emphasizing the balance between annotation efficiency and tracking performance on LaSOT.



**Figure 1.** Comparison of our weakly supervised methods with other fully supervised trackers in terms of annotation costs and performance on LaSOT.

Main contributions of our work are summarized as follows:

- We propose a generalized co-saliency attention mechanism, which can be integrated into various tracking frameworks to achieve weakly supervised tracking. Valuable visual information from unlabeled frames can be explored by our method, which reduces plenty of manual annotation costs during training.
- We record historical visual features and explore their co-salient information via our CSA in the inference. This enables the model to be more robust to challenges like appearance changes and occlusions, which provide a solution for egocentric tracking.
- We integrate our CSA module into two CNN-based methods, and a Transformer-based method. Extensive experiments on four general VOT datasets and an egocentric tracking dataset demonstrate that our weakly supervised tracking models can achieve over 95% performance of full supervision, using only 3.33% manual annotations.

## 2. Related Work

### 2.1. Supervised Visual Tracking

Existing deep learning-based tracking methods can be divided into two categories in terms of the type of backbone networks, which are CNN-based trackers and Transformer-based trackers, respectively. Some representative CNN-based methods include Siamese-based trackers [5,16–19] and discriminative methods [20–25]. Due to the powerful global representation capability of Transformer networks, the recent state-of-the-art (SOTA) tracking methods are gradually transitioning to employ Transformer-based backbone networks. To give full play to the global advantages of Transformer networks, one-stream tracking frameworks based on pure Transformer architectures are proposed, such as OTrack [6] and MixFormer [26]. These methods adopt the pretrained ViT [27] to merge the feature

extraction and feature fusion, which achieves the deep fusion between template and search images. Generalized representations by pretrained models also boost tracking results. Although aforementioned SOTA trackers perform well on many tracking scenes, their models require a large amount of manually annotated video data during the fully supervised training process. Annotating videos frame-by-frame is inevitably a labor- and time-consuming task.

## 2.2. Label-Efficient Visual Tracking

Regarding label-efficient visual tracking, some works aim to achieve unsupervised tracking. UDT [9] proposes an unsupervised tracker trained on large-scale unlabeled videos. This method learns to locate targets by repeatedly performing forward tracking and backward verification. PUL [10] proposes a progressive unsupervised tracking framework. A background discrimination model based on contrastive learning is employed to progressively mine temporal corresponding patches. A noise-robust loss function is further proposed to mitigate the effect from noisy samples. A crop-transform-paste operation is developed in [11], which enriches training data by simulating appearance variations. In its self-supervised tracking setting, only one annotated frame per video is required. Unlike these methods that rely on complex frameworks incompatible with existing advanced trackers, our CSA module can be easily integrated into varying supervised tracking pipelines to enable weakly supervised tracking, offering greater flexibility and practicality. Moreover, due to the lack of precise annotations to guide the training process, the tracking performance of these unsupervised methods still has gaps with supervised trackers.

To make a trade-off between the performance and annotation costs, Liu et al. [12] propose a reliable sample selection strategy for weakly supervised tracking. Given the initial annotation, a correlation filter-based tracker using hand-craft features is employed to generate pseudo-labels for subsequent frames. These pseudo-labels are then used to crop training samples for Siamese-based trackers. Poor-quality samples assessed by the proposed strategy will be filtered. Approximately 9.13% manual annotations are required in this work for weakly supervised training, and the obtained performance is close to 80% of full supervision. This method focuses solely on generating accurate training labels to enable weakly supervised training, with performance upper-bounded by that of fully supervised models. In contrast, our approach integrates the proposed CSA module into existing trackers to enhance the model itself via co-saliency learning, allowing it to surpass the fully supervised performance in some cases. Experiments show that our method can obtain competitive performance with their baseline trackers, using only 3.33% manual annotations for training.

## 2.3. Egocentric Visual Tracking

Since egocentric vision is closer to human perceptions of the real world, it is important for various real-world applications, such as virtual reality [28], surveillance [29], and vision control [30]. As one of the fundamental tasks for egocentric vision, visual object tracking on egocentric sequences has begun to be explored by more and more researchers. Dunnhofer et al. [1] propose an egocentric tracking benchmark, TREK-150. This benchmark consists of 150 densely annotated videos under kitchen scenes. EgoTracks dataset [2] has recently been proposed to further support the development of egocentric tracking tasks. This dataset sources videos from Ego4D [4] and supplements dense annotations for both training and testing subsets.

Compared with the general tracking task (third-person vision), egocentric tracking is more challenging due to rapid appearance changes, large camera motions, occlusions caused by frequent hand interactions, and so on. Since our method can learn co-salient

foreground information from multiple frames. The recorded rich appearance changes enable our model to better adapt to egocentric tracking. To demonstrate this conjecture, we additionally evaluate our method on egocentric tracking dataset TREK-150. More comparisons and analysis can be found in Section 4.3.

### 3. Approach

Algorithm 1 summarizes the weakly supervised training pipeline of our CSA-enhanced tracker. Training videos are segmented using sparse annotations, and missing labels are interpolated. Given a baseline tracker, our CSA modules are integrated into it to achieve weakly supervised training.

---

#### Algorithm 1 Weakly Supervised Training of CSA-Enhanced Tracker

---

- 1: **Input:** Training videos  $V$ , sparse annotations  $A$ , baseline tracker  $T$ ;
  - 2: **Output:** CSA-enhanced Tracker  $T_{CSA}$  with weakly supervised training;
  - 3: **Segment** each video of  $V$  into short fragments based on sparse annotations  $A$ ;
  - 4: **Interpolate** missing annotations for unlabeled frames using linear interpolation;
  - 5: **Integrate** CSA modules into the baseline tracker  $T$  to build  $T_{CSA}$ ;
  - 6: **Train**  $T_{CSA}$  in a weakly supervised manner:
    - Sample *template* and *search image* pairs from annotated frames;
    - Sample *intermediate images* from unannotated frames;
  - 7: **return** CSA-enhanced Tracker  $T_{CSA}$  with weakly supervised training.
- 

In this section, we first introduce our weakly supervised module based on co-saliency learning in Section 3.1. Then, we describe our weakly supervised data processing in Section 3.2. In the end, we present how we integrate our method into existing tracking frameworks to achieve weakly supervised tracking in Section 3.3 (CNN-based framework) and Section 3.4 (Transformer-based framework), respectively.

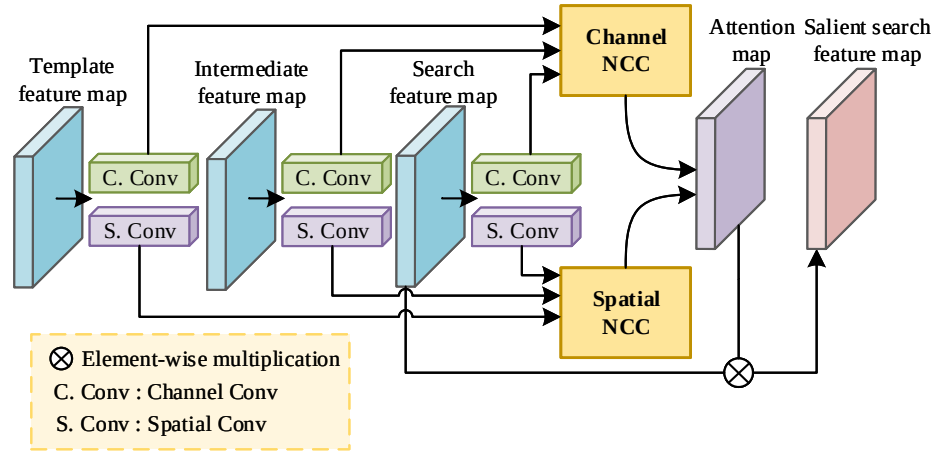
#### 3.1. Weakly Supervised Co-Saliency Learning

Co-saliency learning aims to extract the common significant foreground areas from a set of images. Inspired by the prior work [14], we propose a co-saliency attention mechanism (CSA) to construct a weakly supervised tracking environment. Figure 2 shows the detailed architecture of the proposed CSA module. The input of the module contains feature maps of multiple frames from the same video, denoted as  $\{f_t\}_{t=1}^T$ .  $T$  represents the total number of feature maps and  $f_t \in \mathbb{R}^{C \times H \times W}$  has  $H \times W$  resolution with  $C$  dimensions. Our CSA module consists of two branches, i.e., spatial branch and channel branch. In doing so, the common foreground information of multiple input frames can be obtained considering both spatial positions and common important channels. Specifically, to reduce computations and implement our CSA as a lightweight module, all of the input feature maps first go through the channel and spatial dimension reduction layers (shown as Channel Conv and Spatial Conv in Figure 2). These two dimension reduction layers are both implemented by a convolutional layer followed with ReLU, and output two types of feature maps with low dimensions  $f_t^c \in \mathbb{R}^{C \times H_L \times W_L}$  and  $f_t^s \in \mathbb{R}^{C_L \times H \times W}$ , where  $C_L$ ,  $H_L$  and  $W_L$  are much smaller than  $C$ ,  $H$ , and  $W$ . Then, we employ the normalized cross correlation operation (NCC) to calculate the similarity among multiple frames. The NCC operation of two feature vectors with dimension  $C_c$  can be formulated as

$$NCC(P, Q) = \frac{1}{C_c} \frac{\sum_{g=1}^{C_c} (P_g - \mu_P)(Q_g - \mu_Q)}{\sigma_P \times \sigma_Q}, \quad (1)$$

where  $(\mu_P, \mu_Q)$  and  $(\sigma_P, \sigma_Q)$  represent the couples of the mean and standard deviation of the feature vectors  $P$  and  $Q$ , respectively. The aforementioned two types of low-dimension

feature map sets  $\{f_t^c\}_{t=1}^T$  and  $\{f_t^s\}_{t=1}^T$  are, respectively, input to the channel NCC block and spatial NCC block to calculate the co-salient information of all  $T$  frames in channel and spatial levels.



**Figure 2.** Architecture of the co-saliency learning module. It captures the common salient information of multiple frames in terms of both spatial and channel levels.

In the spatial NCC block, every spatial location  $(i, j)$  ( $1 \leq i \leq H, 1 \leq j \leq W$ ) of the feature map  $f_t^s$  with dimension  $C_L \times H \times W$  can be viewed as a  $C_L$  dimensional feature vector (denoted as  $f_t^s(i, j)$ ) that describes the local information at  $(i, j)$ . We employ NCC to calculate the similarity of the information in the frame's position  $(i, j)$  with the information in all positions of other frames, formulated as

$$\begin{aligned} SA_t^{i,j} &= NCC(f_t^s(i, j), f_k^s(h, w)), \\ 1 &\leq k \leq T, k \neq t \\ 1 &\leq h \leq H \\ 1 &\leq w \leq W. \end{aligned} \quad (2)$$

The output of the spatial NCC block is a 3D attention map  $SA_t \in \mathbb{R}^{(T-1)HW \times H \times W}$ . It records the correlation information between each position of the frame  $t$  and all positions of other  $(T - 1)$  frames. Thus, spatial locations of the common foreground consulting with multiple frames are activated.

The process of the channel NCC block is similar to the spatial NCC block. First, the feature map  $f_t^c$  with dimension  $C \times H_L \times W_L$  is flattened to 2D matrices. It represents that every channel in the feature can be viewed as a  $H_L W_L$  dimensional vector  $f_t^c(l)$ . The channel NCC block computes the channel attention map  $CA_t \in \mathbb{R}^{(T-1)C \times C}$ , which records the similarity between each channel of the frame  $t$  and all channels of other  $(T - 1)$  frames. The common important channels of all video frames can then be enhanced. This process can be formulated as

$$\begin{aligned} CA_t^l &= NCC(f_t^c(l), f_k^c(e)), \\ 1 &\leq k \leq T, k \neq t \\ 1 &\leq e \leq C. \end{aligned} \quad (3)$$

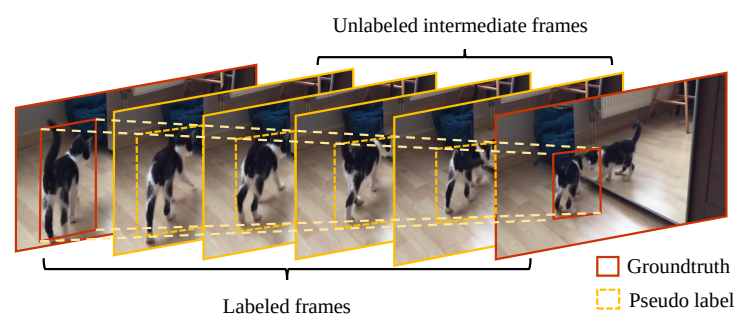
The two attention maps recording the similarity information, i.e.,  $SA$  and  $CA$ , are then summarized by a convolutional layer, and transformed to  $Z_s \in \mathbb{R}^{1 \times H \times W}$  and  $Z_c \in \mathbb{R}^{C \times 1 \times 1}$ , respectively. The final spatial-channel attention map is formulated as follows:

$$Attention = \text{sigmoid}(Z_c \otimes Z_s). \quad (4)$$

Among them, *Attention* is a  $C \times H \times W$  matrix, where common important channels and the co-salient spatial positions are highlighted with reference to multiple video frames, while other information will be weakened. In this paper, we calculate the co-saliency attention map of the current frame's feature map, with reference to the information from the other two frames, including the initial template and a random intermediate frame. The attention map is multiplied with the feature map of the current frame, making the target position and important channels more salient. Meanwhile, the background information and other trivial channels of the final feature map of the current frame are suppressed.

### 3.2. Weakly Supervised Data Processing

Since the target positions and shapes usually do not change much in a 30-frame clip, which occurs in a second for most videos, we adopt sparsely labeled training datasets to achieve weakly supervised tracking. That is, labeling one bounding box of the target every 30 frames. The input images to CSA are used to generate co-saliency attention maps, which do not require precise target annotations but only need the target to be present within the search regions cropped based on pseudo-labels. As a result, CSA inputs from unlabeled frames are highly tolerant to pseudo-label inaccuracies. Therefore, in this work, we employ linear interpolation to generate pseudo-labels for unlabeled frames, illustrated as Figure 3.



**Figure 3.** Illustration of our weakly supervised data processing. Pseudo-labels of unlabeled frames are generated based on the groundtruth of the first and last frames from one short fragment in a linear interpolation manner.

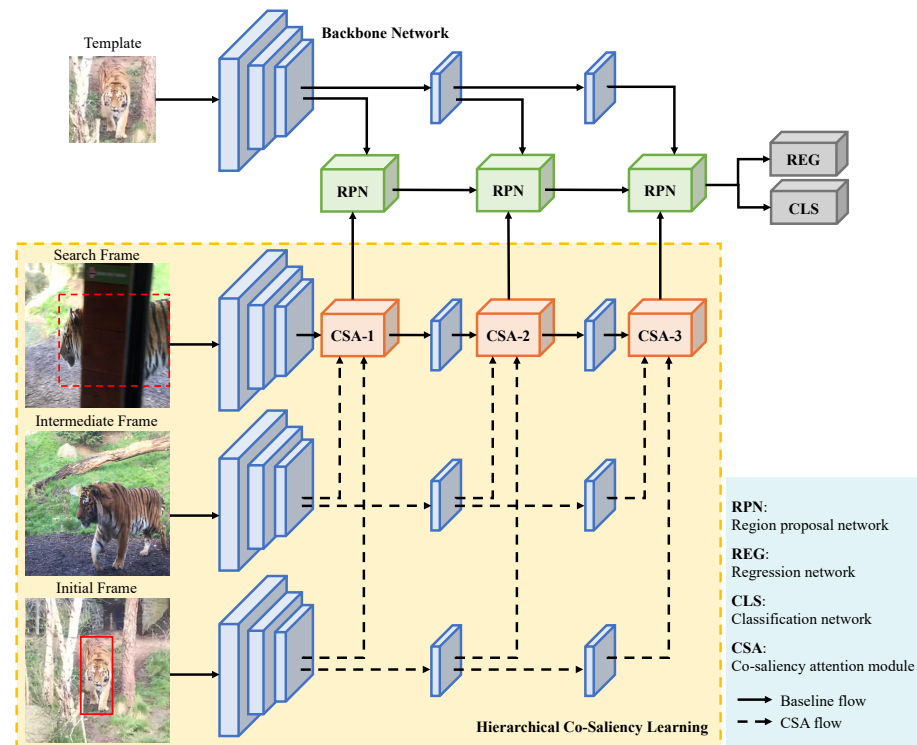
Although linear interpolation is not reliable for modeling complex scenarios such as nonlinear target motion, it is sufficient for our method because precise annotations are not strictly required. To support this claim, we evaluate the quality of the pseudo-labels generated on LaSOT dataset using two metrics. As shown in Table 1, we report results under varying annotation sparsity. “IoU” denotes the Intersection over Union between pseudo-labels and groundtruth annotations, while “Target ratio” measures the proportion of the target included in the search region cropped based on the pseudo-label. We observe that while the IoU between pseudo-labels and groundtruth drops significantly with increased sparsity (e.g., from 0.679 at sparsity 30 to 0.445 at sparsity 150), the target ratio remains consistently high. Even at a sparsity level of 150, the target ratio exceeds 0.9, indicating that the cropped search regions still contain nearly complete target information. This property fits well with our method, as the CSA module only requires the target to be present in the search region, not precisely localized. Thus, pseudo-labels from linear interpolation are sufficiently reliable, enabling effective weakly supervised training.

**Table 1.** Quality evaluation of the generated pseudo-labels on LaSOT dataset with different sparseness.

| Sparseness   | 30    | 60    | 90    | 150   |
|--------------|-------|-------|-------|-------|
| IoU          | 0.679 | 0.575 | 0.516 | 0.445 |
| Target ratio | 0.968 | 0.945 | 0.926 | 0.907 |

### 3.3. CNN-Based Weakly Supervised Tracking

Taking the baseline tracker SiamRPN++ [5] as an example, we present how we hierarchically integrate our CSA module into a CNN-based tracking framework to achieve weakly supervised training, shown as Figure 4.



**Figure 4.** Framework of our CNN-based weakly supervised tracking method. We choose SiamRPN++ as the baseline model. The backbone network, RPN, REG, and CLS blocks maintain the same configurations as the baseline.

For the baseline tracker SiamRPN++, its inputs consist of an initial template image  $z$  and a current search image  $x_c$ . A CNN-based backbone network is employed to extract their multi-level feature maps, and then places them into corresponding region proposal networks (RPN blocks in Figure 4). After fusing the outputs from multiple RPN blocks, the classification and regression branches will give a dense prediction of the target. The former is in charge of choosing the best proposal through classifying the foreground and background, while the latter is used to refine the proposal and estimate the offsets of anchor boxes. This process can be formulated as

$$\begin{aligned} A_{w \times h \times 2k}^{cls} &= [\varphi(x_c)]_{cls} \star [\varphi(z)]_{cls} \\ A_{w \times h \times 4k}^{reg} &= [\varphi(x_c)]_{reg} \star [\varphi(z)]_{reg}, \end{aligned} \quad (5)$$

where  $\varphi(\cdot)$  represents the feature maps extracted from the backbone network,  $k$  represents the number of predefined anchors, and  $\star$  is a convolution operation on feature maps of search regions, with the template features as the convolution kernel.

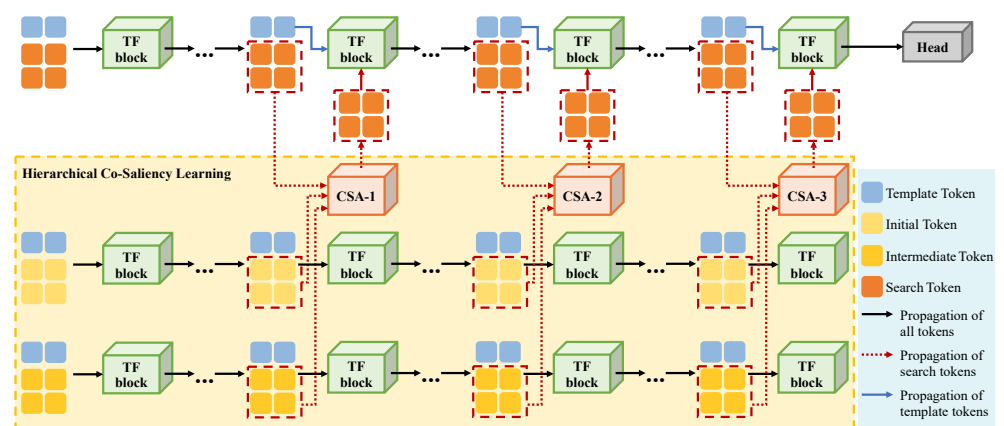
Different from the baseline, there are three different search regions in the inputs of our model, which are cropped from the initial frame, an unlabeled intermediate frame, and the current search frame, respectively, namely  $x_t$ ,  $x_m$ , and  $x_c$ . It is worth noting that intermediate frames are cropped based on generated pseudo-labels. Each of these three search regions has a shared search branch to extract feature maps. As shown in Figure 4, multiple of our CSA modules are hierarchically inserted after different layers of the backbone network. The features of the corresponding layers output by the three search branches are input to our CSA modules. For the current search branch, the input of the next backbone layer will be replaced by the output of the corresponding CSA module, i.e., the enhanced feature maps of the current search image. After integrating our CSAs, the tracking process can be reformulated as

$$\begin{aligned} A_{w \times h \times 2k}^{cls} &= [CSA(\varphi(x_t), \varphi(x_m), \varphi(x_c))]_{cls} \star [\varphi(z)]_{cls} \\ A_{w \times h \times 4k}^{reg} &= [CSA(\varphi(x_t), \varphi(x_m), \varphi(x_c))]_{reg} \star [\varphi(z)]_{reg}. \end{aligned} \quad (6)$$

We can find that after integrating our CSA modules, the tracking framework is able to use both labeled (i.e., the initial and current search frames) and unlabeled data (i.e., intermediate frames), therefore constructing a weakly supervised training environment. The precisely labeled data can provide accurate target information and guide the model optimization by calculating precise losses, while the unlabeled data is mainly utilized to calculate the co-saliency information of multiple frames, and then enhance the current search features.

### 3.4. Transformer-Based Weakly Supervised Tracking

Due to their powerful global representation ability, Transformer-based tracking methods have gradually become the mainstream. To demonstrate the generalization ability of our method on Transformer-based tracking frameworks, we choose a one-stream Transformer-based tracker OTrack [6] as the baseline, and integrate our CSA modules into its framework. In OTrack, the template and search region are first split to multiple patches with a fixed size. Each patch can be treated as a processing unit in the Transformer network, namely token. Passing through a patch embedding layer, template tokens and search tokens are concatenated together and iteratively input into multiple Transformer blocks (TF blocks), shown as the top row in Figure 5. After executing self-attentions and cross-attentions in these TF blocks, the search tokens output from the last TF block are sent to a head module to predict the target state.



**Figure 5.** Framework of our Transformer-based weakly supervised tracking method. We choose OTrack as the baseline model. As the same as the baseline, the model ViT-B [27] consisting of 12 sequential Transformer blocks (TF block) is employed as the backbone network.

Similar to CNN-based integration, we employ three shared backbone branches to encode three types of tokens along with the template tokens, including initial tokens, intermediate tokens, and search tokens. These tokens are processed through a series of TF blocks, and three CSA modules are integrated after specific TF blocks to facilitate information interaction and enhancement among the three branches. Specifically, the output tokens from the current TF blocks in each of the three branches, including initial tokens, intermediate tokens, and search tokens, are reshaped into feature maps. These feature maps are then passed to the corresponding CSA modules for co-saliency calculation. The CSA module enhances the search tokens by incorporating additional context and information from the other branches. The enhanced search tokens will then replace the original search tokens, continuing the Transformer encoding process with the updated tokens, as shown by the orange tokens in Figure 5.

## 4. Experiments

Our work is implemented in Python using PyTorch. Training is performed using NVIDIA A800 GPUs, Santa Clara, CA, USA, while inference is conducted on a single NVIDIA GeForce RTX 2080 Ti GPU, Santa Clara, CA, USA. To show the generalization ability of our weakly supervised CSA module, we choose two classic CNN-based trackers (i.e., an anchor-based tracker SiamRPN++ [5] and an anchor-free tracker Ocean [16]) and a Transformer-based tracker OTrack [6] as baselines. We transform them into weakly supervised tracking algorithms by integrating our CSA module, denoted as SiamRPN++<sub>CSA</sub>, Ocean<sub>CSA</sub>, and OTrack<sub>CSA</sub>, respectively. We compare the tracking performance of our weakly supervised methods with the three baseline trackers and other trackers on four general VOT benchmarks, including LaSOT [7], GOT-10k [8], OTB2015 [31], and UAV123 [32]. In addition, to further demonstrate the effectiveness of our method on egocentric tracking, we also evaluate our method OTrack<sub>CSA</sub> on an egocentric tracking benchmark, TREK-150 [1].

### 4.1. Implementation Details

**Network architecture.** All of our methods follows the same input size as their baselines. For SiamRPN++<sub>CSA</sub> and Ocean<sub>CSA</sub>, the sizes of the input template images and search regions are set to  $127 \times 127$  and  $255 \times 255$ , while for OTrack<sub>CSA</sub>, they are set to  $128 \times 128$  and  $256 \times 256$ . Regarding our CSA module, the spatial dimension reduction layer is implemented by a convolutional layer where the kernel size is 1. Its output dimension is set to 128 for CNN and 256 for Transformer. Meanwhile, the channel dimension reduction layer is a convolutional layer where the kernel size is 7, the stride is 2, and the output dimension is the same as the input. For CNN-based methods, we choose ResNet50 [33] as the backbone, and CSA modules are integrated on the last three layers (since Ocean does not apply multi-layer features to calculate cross-correlation, we only set up one CSA module on its last layer). For Transformer-based methods, we choose ViT-B [27] as the backbone, and similarly set up our CSA modules on the last three Transformer blocks.

**Training.** We follow the same training settings of corresponding baseline trackers. For both two CNN-based methods, we use synchronized stochastic gradient descent (SGD) over 8 GPUs where the batch size of each GPU is 16. GOT-10k [8], LaSOT [7], ImageNet VID [34], and TrackingNet [35] are applied to train our CNN-based models. For Transformer-based method, we use AdamW optimizer over 4 GPUs where batch size of each GPU is 32. We follow the three video datasets in the baseline training dataset settings to train our model, i.e., GOT-10k, LaSOT, and TrackingNet. It is worth noting that we use sparse annotations for all training datasets, i.e., using one annotation per 30 frames. Other unlabeled frames

are arranged pseudo-labels and sampled only as intermediate frames without involving tracking loss calculation.

**Inference.** In the inference, we set up a memory pool to record feature maps of historical search regions. It consists of the feature map of the initial frame's search region, and a real-time updated feature of an historical frame with the highest confidence score. This confidence score considers both the confidence of the tracking result and the time factor. It can be formulated as

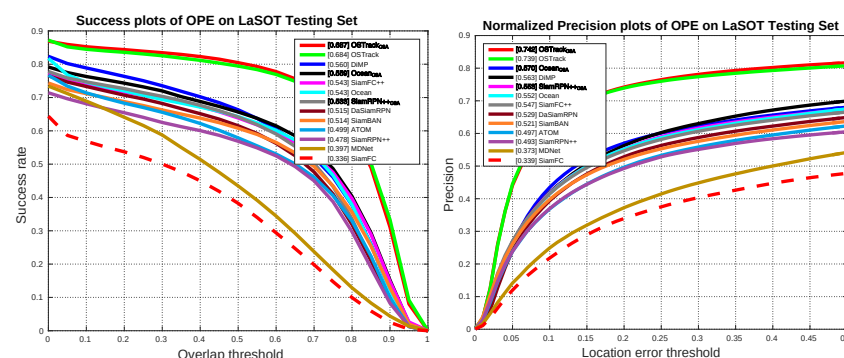
$$S_{penalty}^t = Conf_t + \frac{\lg N_t}{1000}, \quad (7)$$

where  $Conf_t$  and  $N_t$  represent the confidence score of the tracking result and the frame number, respectively. It is worth noting that  $Conf_t$  is generated from tracking models along with the tracking results.

#### 4.2. Performance Comparison on General VOT

We compare our three weakly supervised methods with fully supervised baselines and other trackers on four general VOT benchmarks, i.e., LaSOT, OTB2015, UAV123, and GOT-10k. Moreover, we also compare with existing weakly supervised and unsupervised trackers on OTB2015 and UAV123.

**LaSOT.** It is a large-scale and long-term tracking dataset containing 1120 training sequences and 280 testing sequences with 70 object categories. The average sequence length is more than 2500 frames and there are more than 3.5 M frames in total. Sequences in LaSOT often occur in the situation that targets disappear and re-appear in the view, which better reflects the practical performance of trackers in the wild. The benchmark uses precision (Pre.), normalized precision (Norm Pre.), and success (Succ.) following one-pass evaluation (OPE) to measure the performance of trackers. The precision compares the distance between the tracking results and groundtruth in pixels to measure center location error. To make the evaluation robust to the target size and image resolution, the normalized precision is used to rank trackers using Area Under Curve (AUC) between 0 to 0.5. In addition, the success calculates the Intersection over Union (IoU) between the tracking results and groundtruth to reflect the accuracy of the target's position and scale. Figure 6 shows the tracking results of our three weakly supervised methods (denoted as the subscript “CSA”) and 10 fully supervised trackers, including SiamFC [17], MDNet [20], ATOM [22], SiamBAN [36], DaSiamRPN [37], SiamFC++ [38], DiMP [21], SiamRPN++ [5], Ocean [16], and OTrack [6]. Our weakly supervised methods achieve competitive performance to SOTA fully supervised trackers. Using only 3.33% manual annotations in training, our CSA can further improve three baseline trackers by averaged 2.5% in terms of Succ.



**Figure 6.** Success and norm precision plots on LaSOT dataset. Our weakly supervised methods obtain comparable results to their fully supervised baselines.

**OTB2015.** It is a classic short-term tracking dataset that contains 100 sequences. Two metrics, precision and success, are used for evaluation. As shown in Table 2, our method, OTrack<sub>CSA</sub>, improves its fully supervised baseline by 1.7% Succ. and 3.1% Pre. In addition, our methods achieve comparable tracking performance with fully supervised trackers, and are superior to existing weakly supervised trackers. Regarding unsupervised methods, we can find that they still have some performance gaps with SOTA trackers.

**UAV123.** Unlike the above datasets, it captures 123 videos via drone aerial photography with more than 110,000 frames. It also uses the precision and success metrics to evaluate the tracking results. As shown in Table 2, our CSA boosts three baseline trackers by 2.0% Pre. on average, using only one-thirtieth of manual annotations.

**Table 2.** Performance comparison on OTB2015 and UAV123. Fully supervised, weakly supervised, and unsupervised trackers are denoted as white, yellow, and pink. The top three results are marked as red, blue, and green, respectively.

|                          | OTB2015      |              |                          | UAV123       |              |
|--------------------------|--------------|--------------|--------------------------|--------------|--------------|
|                          | Succ.        | Pre.         |                          | Succ.        | Pre.         |
| MDNet [20]               | 0.652        | 0.878        | SiamRPN [18]             | 0.527        | 0.748        |
| Ocean [16]               | 0.658        | 0.863        | MDNet [20]               | 0.528        | 0.747        |
| ATOM [22]                | 0.665        | 0.870        | DaSiamRPN [37]           | 0.586        | 0.796        |
| DAT [39]                 | 0.668        | 0.895        | Ocean [16]               | 0.596        | 0.798        |
| OTrack [6]               | 0.675        | 0.884        | SiamRPN++ [5]            | 0.606        | 0.797        |
| VITAL [40]               | 0.682        | <b>0.917</b> | SiamCAR [41]             | 0.614        | 0.760        |
| ECO [42]                 | <b>0.691</b> | 0.910        | ATOM [22]                | <b>0.631</b> | <b>0.843</b> |
| SiamRPN++ [5]            | <b>0.693</b> | 0.897        | OTrack [6]               | <b>0.678</b> | <b>0.883</b> |
| RSS [12]                 | 0.675        | 0.779        | RSS [12]                 | -            | -            |
| SS-annotated [11]        | 0.634        | -            | SS-annotated [11]        | 0.521        | -            |
| Ocean <sub>CSA</sub>     | 0.676        | 0.900        | SiamRPN++ <sub>CSA</sub> | 0.616        | 0.819        |
| SiamRPN++ <sub>CSA</sub> | 0.690        | <b>0.916</b> | Ocean <sub>CSA</sub>     | 0.624        | 0.834        |
| OTrack <sub>CSA</sub>    | <b>0.692</b> | <b>0.915</b> | OTrack <sub>CSA</sub>    | <b>0.677</b> | <b>0.884</b> |
| ResPUL [10]              | 0.584        | -            | ResPUL [10]              | -            | -            |
| UDT [9]                  | 0.594        | 0.760        | UDT [9]                  | -            | -            |
| LU DT [43]               | 0.602        | 0.769        | LU DT [9]                | -            | -            |
| SS-detector [11]         | 0.619        | -            | SS-detector [11]         | 0.519        | -            |

**GOT-10k.** It contains over 10,000 video segments with 563 object classes and 87 motion forms for short-term tracking. The testing set consists of 180 videos with 150 object classes and 15 motion classes. Different from LaSOT dataset, it introduces average overlap (AO) and success rate (SR) as indicators. The former denotes the average of overlaps between the estimated bounding boxes and groundtruth, while the latter measures the percentage of successfully tracked frames where the overlaps exceed a threshold. The threshold is set to 0.5 and 0.75, and the corresponding indicators are denoted as SR<sub>0.5</sub> and SR<sub>0.75</sub>, respectively. We compare our methods with 10 fully supervised trackers, as shown in Table 3. Our CSA yields SiamRPN++ 3.7% and 5.6% improvements in terms of the AO and SR<sub>0.5</sub>. Meanwhile, Ocean\_CSA increases SR<sub>0.75</sub> by 4.2%. Regarding the speed, since we record features of historical frames by a memory pool in the inference, tracking speeds of our methods are not sacrificed much, which can still achieve real-time tracking.

**Table 3.** Comparison of the tracking results and speeds on GOT-10k dataset.

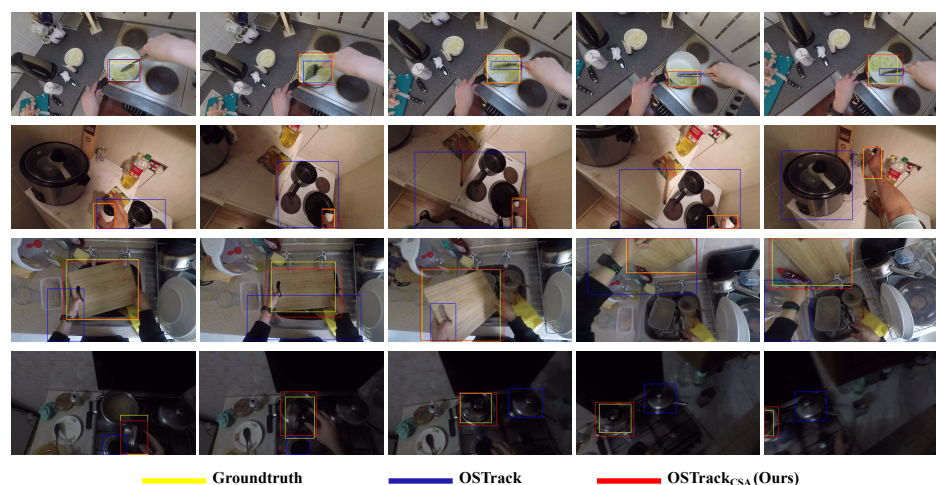
|                          | AO           | SR <sub>0.5</sub> | SR <sub>0.75</sub> | FPS |
|--------------------------|--------------|-------------------|--------------------|-----|
| SiamRPN++ [5]            | 0.517        | 0.616             | 0.325              | 45  |
| ATOM [22]                | 0.556        | 0.634             | 0.402              | 30  |
| SiamCAR [41]             | 0.569        | 0.670             | 0.415              | 52  |
| Ocean [16]               | 0.590        | 0.696             | 0.438              | 51  |
| SiamFC++ [38]            | 0.595        | 0.695             | 0.479              | 90  |
| DiMP [21]                | 0.611        | 0.717             | 0.492              | 43  |
| PrDiMP [23]              | <b>0.634</b> | <b>0.738</b>      | <b>0.543</b>       | 30  |
| OSTrack [6]              | <b>0.717</b> | <b>0.809</b>      | <b>0.685</b>       | 89  |
| SiamRPN++ <sub>CSA</sub> | 0.554        | 0.672             | 0.369              | 36  |
| Ocean <sub>CSA</sub>     | 0.611        | 0.707             | 0.480              | 46  |
| OSTrack <sub>CSA</sub>   | <b>0.684</b> | <b>0.789</b>      | <b>0.634</b>       | 65  |

#### 4.3. Performance Comparison on Egocentric Tracking

To demonstrate our conjecture that our CSA can enable the tracking model to better adapt to egocentric tracking, we evaluate our superior method OSTRack<sub>CSA</sub> on the egocentric tracking benchmark, TREK-150 (Figure 7) [1]. It consists of 150 sequences in kitchen scenes. All of them are shot under first-person vision. As shown in Table 4, we present the performance of the top ten methods evaluated in its official paper, and the baseline OSTRack. It is worth noting that all of the presented methods are fully supervised trackers except for our method OSTRack<sub>CSA</sub>. It shows that using only 3.33% of manual annotations for training, our method obtains significant performance improvement by 3.6% Succ., and is superior than the prior SOTA method LTMU-H by 7.7% Succ.

**Table 4.** Performance comparison on egocentric dataset TREK-150.

|                        | Succ.        | Norm Pre.    |
|------------------------|--------------|--------------|
| PrDiMP [23]            | 0.388        | 0.402        |
| ECO [42]               | 0.388        | 0.413        |
| ATOM [22]              | 0.400        | 0.420        |
| TRASFUST [44]          | 0.403        | 0.419        |
| LTMU [45]              | 0.437        | 0.448        |
| LTMU-F [1]             | 0.456        | 0.477        |
| LTMU-H [1]             | 0.461        | 0.486        |
| OSTrack [6]            | 0.502        | 0.562        |
| OSTrack <sub>CSA</sub> | <b>0.538</b> | <b>0.602</b> |

**Figure 7.** Qualitative comparisons of our method with its fully supervised baseline on TREK-150.

In addition, we illustrate qualitative comparisons of our weakly supervised method with its fully supervised baseline tracker on TREK-150. We can find that our method can better address difficult challenges, such as scale variations, appearance changes, partial occlusions, and fast motion.

#### 4.4. Ablation Study

To deeply analyze our method from different aspects, we execute a set of ablation studies on LaSOT.

**Impact of CSA.** To demonstrate the effectiveness of our CSA, we present tracking results of three variants of baseline trackers Ocean and OTrack, shown as Table 5. Among them, “Fully S.” represents the original fully supervised baseline, “Weakly S.” represents using 3.33% manual annotations to train the baseline model, and “CSA” represents our weakly supervised method. The tracking performance inevitably decreases when reducing the number of annotations for training, but this degradation can be significantly addressed by our CSA method. Since our CSA can calculate co-salient attentions of multiple frames, valuable visual information in unlabeled frames can be captured and utilized by our tracking model.

**Table 5.** Impact of our CSA module on the tracking performance.

|           | Fully S. | Ocean<br>Weakly S. | CSA          | Fully S. | OTrack<br>Weakly S. | CSA          |
|-----------|----------|--------------------|--------------|----------|---------------------|--------------|
| Succ.     | 0.543    | 0.532              | <b>0.559</b> | 0.684    | 0.676               | <b>0.687</b> |
| Norm Pre. | 0.637    | 0.628              | <b>0.643</b> | 0.777    | 0.768               | <b>0.788</b> |
| Pre.      | 0.552    | 0.551              | <b>0.570</b> | 0.739    | 0.736               | <b>0.742</b> |

**Amount of annotations.** We set different label sparseness of training datasets for our method, SiamRPN++<sub>CSA</sub>. As shown in Table 6, our method is tolerant of a wide range of annotation sparseness. The tracking performance of our method is still superior to its fully supervised baseline even when the amount of annotations is reduced by 150 times.

**Table 6.** Impact of the sparseness degree of manual annotations in the training dataset.

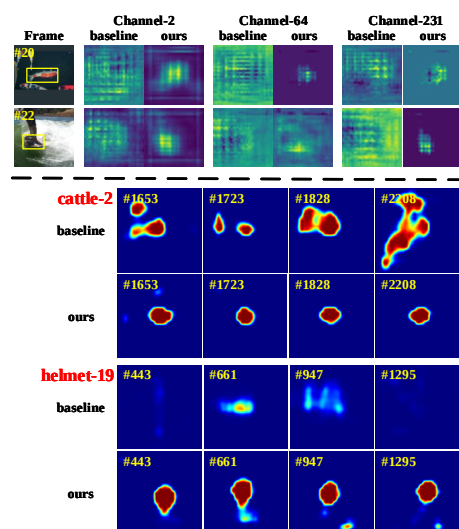
| Tracker                                 | Succ.        | Norm Pre.    | Pre.         |
|-----------------------------------------|--------------|--------------|--------------|
| SiamRPN++ <sup>1</sup>                  | 0.478        | 0.572        | 0.493        |
| SiamRPN++ <sub>CSA</sub> <sup>30</sup>  | <b>0.533</b> | <b>0.637</b> | <b>0.553</b> |
| SiamRPN++ <sub>CSA</sub> <sup>90</sup>  | 0.519        | 0.624        | 0.538        |
| SiamRPN++ <sub>CSA</sub> <sup>150</sup> | 0.497        | 0.600        | 0.504        |

**Number of reference frames in CSA.** To explore the impact of our CSA with different numbers of reference frames, we evaluate several CSA variants, shown as Table 7. Numbers of reference frames are denoted as  $T$ . We observe that our method reaches peak performance at  $T = 3$ , and the speed does not decrease much. Therefore,  $T = 3$  is our default setting in this paper. There are two potential reasons for the performance degradation when  $T = 4$ . First, high-confidence reference frames often occur close in time, limiting the diversity of target appearances. Second, maintaining a larger number of frames increases the risk of error accumulation, which may negatively affect tracking quality.

**Table 7.** Impact of the number of reference frames in our CSA.

| Tracker                                 | Succ.        | Norm Pre.    | Pre.         | FPS |
|-----------------------------------------|--------------|--------------|--------------|-----|
| SiamRPN++ <sup>T=2</sup> <sub>CSA</sub> | 0.485        | 0.584        | 0.513        | 38  |
| SiamRPN++ <sup>T=3</sup> <sub>CSA</sub> | <b>0.533</b> | <b>0.637</b> | <b>0.553</b> | 36  |
| SiamRPN++ <sup>T=4</sup> <sub>CSA</sub> | 0.508        | 0.606        | 0.529        | 34  |

**Qualitative analysis.** To demonstrate how our method improves tracking performance over the baseline SiamRPN++, we visualize the response maps and search region features in Figure 8. Specifically, we compare feature maps from layer-5 of the backbone, illustrating three example channels. Our method produces clearer and more target-focused features, with reduced background noise. Different channels capture different aspects of the target, contributing to richer representations. As a result, the classification branch generates more accurate and robust response maps. Figure 8 compares two sequences: “cattle-2” with distractors and “helmet-19” with occlusion. Our method handles both challenges better than the baseline. Overall, the CSA module enhances target features using multi-frame context, leading to more precise localization and improved tracking performance.

**Figure 8.** Visualization of search regions’ feature maps and response maps generated from the baseline SiamRPN++ and our method SiamRPN++<sub>CSA</sub>.

## 5. Conclusions

In this paper, to achieve label-efficient but well-performed tracking, we propose a generalized co-saliency attention mechanism, which can be hierarchically integrated into various tracking frameworks and enable the model to be trained in a weakly supervised manner. The proposed method aims to capture the common significant targets in multiple search regions cropped from different frames, from both spatial and channel levels. The target representation in the current search image can be enhanced via the output co-salient attention map. To achieve weakly supervised training, we use sparsely labeled training data, and make full use of remaining unlabeled frames by generating pseudo-labels. To demonstrate the effectiveness and generalization ability of our method, we integrate our CSA module into three different tracking frameworks. Extensive experiments on four general VOT datasets and an egocentric tracking dataset show that our methods can obtain comparable tracking performance to other fully supervised trackers using only one-thirtieth manual annotations for training. Our method also provides a solution for egocentric tracking, which obtains the the new SOTA performance on TREK-150.

### Limitations

During training, extreme fast motion may lead to inaccurate pseudo-labels. While at test time, fast motion or occlusion can cause the target to disappear from the search region, introducing noise into co-saliency learning and increasing tracking failure risk. In the future, incorporating robust target modeling, such as 3D Gaussian distributions or motion-aware interpolation, could enhance reliability and accuracy.

**Author Contributions:** All authors contributed to the idea for the article; Y.G., C.B. and D.W. provided the initial idea and framework regarding this manuscript; the literature search and analysis were performed by J.Z.; J.Z. conducted experiments and prepared the manuscript initially. All authors participated in the discussion and modification of the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundations of China grant number (62402084), Basic Scientific Research Funding of the Central Universities of China grant number (DUTZD25225), Liaoning Provincial Science and Technology Joint Program Project grant number (2024011188-JH2/1026), and China Postdoctoral Science Foundation grant number (2024M750319).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Acknowledgments:** The authors express their gratitude to the anonymous reviewers and the editor.

**Conflicts of Interest:** Author Ying Gao was employed by the company Henan Water & Power Engineering Consulting Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

### References

1. Dunnhofer, M.; Furnari, A.; Farinella, G.M.; Micheloni, C. Is first-person vision challenging for object tracking? In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 2698–2710.
2. Tang, H.; Liang, K.J.; Grauman, K.; Feiszli, M.; Wang, W. Egotracks: A long-term egocentric visual object tracking dataset. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, 10–16 December 2023*; Curran Associates: New York, NY, USA, 2023.
3. Zhao, J.; Dai, K.; Zhang, P.; Wang, D.; Lu, H. Robust online tracking with meta-updater. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 6168–6182. [[CrossRef](#)] [[PubMed](#)]
4. Grauman, K.; Westbury, A.; Byrne, E.; Chavis, Z.; Furnari, A.; Girdhar, R.; Hamburger, J.; Jiang, H.; Liu, M.; Liu, X.; et al. Ego4d: Around the world in 3,000 hours of egocentric video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 18995–19012.
5. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
6. Ye, B.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Joint feature learning and relation modeling for tracking: A one-stream framework. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 341–357.
7. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5374–5383.
8. Huang, L.; Zhao, X.; Huang, K. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1562–1577. [[CrossRef](#)]
9. Wang, N.; Song, Y.; Ma, C.; Zhou, W.; Liu, W.; Li, H. Unsupervised deep tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1308–1317.
10. Wu, Q.; Wan, J.; Chan, A.B. Progressive unsupervised learning for visual object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2993–3002.
11. Li, X.; Pei, W.; Wang, Y.; He, Z.; Lu, H.; Yang, M.H. Self-supervised tracking via target-aware data synthesis. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 9186–9197. [[CrossRef](#)]

12. Liu, S.; Xu, X.; Zhang, Y.; Muhammad, K.; Fu, W. A reliable sample selection strategy for weakly supervised visual tracking. *IEEE Trans. Reliab.* **2023**, *72*, 15–26. [\[CrossRef\]](#)
13. Jia, N.; Liu, X.; Sun, Y. Salient Object Detection Network with Selection Mechanism for Consumer Electronics Imaging Systems. *IEEE Trans. Consum. Electron.* **2023**, *70*, 3403–3413. [\[CrossRef\]](#)
14. Liu, J.; Zhu, X.; Zha, Z.J.; Jiang, N. Co-Saliency Spatio-Temporal Interaction Network for Person Re-Identification in Videos. *arXiv* **2020**, arXiv:2004.04979.
15. Wei, W.; Mumtaz, S.; Shah, S.H.; Wang, W. Guest Editorial Artificial Intelligence-Empowered Internet of Things for Human-Centric Consumer Applications. *IEEE Trans. Consum. Electron.* **2024**, *70*, 849–853. [\[CrossRef\]](#)
16. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-aware Anchor-free Tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 771–787.
17. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.
18. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
19. Dai, K.; Wang, D.; Lu, H.; Sun, C.; Li, J. Visual Tracking via Adaptive Spatially-Regularized Correlation Filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4670–4679.
20. Nam, H.; Han, B. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
21. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Learning discriminative model prediction for tracking. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6182–6191.
22. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate tracking by overlap maximization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4660–4669.
23. Danelljan, M.; Gool, L.V.; Timofte, R. Probabilistic regression for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7183–7192.
24. Zhao, J.; Zhang, J.; Li, D.; Wang, D. Vision-based anti-uav detection and tracking. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25323–25334. [\[CrossRef\]](#)
25. Liu, C.; Yuan, Y.; Chen, X.; Lu, H.; Wang, D. Spatial-temporal initialization dilemma: Towards realistic visual tracking. *Vis. Intell.* **2024**, *2*, 35. [\[CrossRef\]](#)
26. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. MixFormer: End-to-End Tracking with Iterative Mixed Attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13608–13618.
27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representation, Online, 3–7 May 2021.
28. Wu, C.K.; Cheng, C.T.; Uwate, Y.; Chen, G.; Mumtaz, S.; Tsang, K.F. State-of-the-art and research opportunities for next-generation consumer electronics. *IEEE Trans. Consum. Electron.* **2022**, *69*, 937–948. [\[CrossRef\]](#)
29. Wang, J.; Zhang, Z.; Li, B.; Lee, S.; Sherratt, R.S. An enhanced fall detection system for elderly person monitoring using consumer home networks. *IEEE Trans. Consum. Electron.* **2014**, *60*, 23–29. [\[CrossRef\]](#)
30. Lee, D.; Park, Y. Vision-based remote control system by motion detection and open finger counting. *IEEE Trans. Consum. Electron.* **2009**, *55*, 2308–2313. [\[CrossRef\]](#)
31. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [\[CrossRef\]](#)
32. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for UAV tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
34. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [\[CrossRef\]](#)
35. Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 300–317.
36. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese box adaptive network for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6668–6677.
37. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 101–117.
38. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12549–12556. [\[CrossRef\]](#)

39. Pu, S.; Song, Y.; Ma, C.; Zhang, H.; Yang, M.H. Deep attentive tracking via reciprocative learning. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018; pp. 1931–1941.
40. Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.W.; Yang, M.H. VITAL: Visual tracking via adversarial learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8990–8999.
41. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6269–6277.
42. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. ECO: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
43. Wang, N.; Zhou, W.; Song, Y.; Ma, C.; Liu, W.; Li, H. Unsupervised deep representation learning for real-time tracking. *Int. J. Comput. Vis.* **2021**, *129*, 400–418. [[CrossRef](#)]
44. Dunnhofer, M.; Martinel, N.; Micheloni, C. Tracking-by-trackers with a distilled and reinforced model. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November–4 December 2020.
45. Dai, K.; Zhang, Y.; Wang, D.; Li, J.; Lu, H.; Yang, X. High-performance long-term tracking with meta-updater. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6298–6307.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.