Sequential Fusion Based Multi-Granularity Consistency for Space-Time Transformer Tracking

Kun Hu[1*], **Wenjing Yang**[1*], **Wanrong Huang**[1], **Xianchen Zhou**[2], **Mingyu Cao**[1], **Jing Ren**[1], **Huibin Tan**[1†]

[1]Department of Intelligent Data Science, College of Computer Science and Technology,
National University of Defense Technology.
[2]College of Sciences, National University of Defense Technology.
hu_kun_@outlook.com, {wenjing.yang, huangwanrong12, zhouxianchen13, caomy720, renjing, tanhb_}@nudt.edu.cn

Abstract

Regarded as a template-matching task for a long time, visual object tracking has witnessed significant progress in space-wise exploration. However, since tracking is performed on videos with substantial time-wise information, it is important to simultaneously mine the temporal contexts which have not yet been deeply explored. Previous supervised works mostly consider template reform as the breakthrough point, but they are often limited by additional computational burdens or the quality of chosen templates. To address this issue, we propose a Space-Time Consistent Transformer Tracker (STCFormer), which uses a sequential fusion framework with multi-granularity consistency constraints to learn spatiotemporal context information. We design a sequential fusion framework that recombines template and search images based on tracking results from chronological frames, fusing updated tracking states in training. To further overcome the over-reliance on the fixed template without increasing computational complexity, we design three space-time consistent constraints: Label Consistency Loss (LCL) for label-level consistency, Attention Consistency Loss (ACL) for patch-level ROI consistency, and Semantic Consistency Loss (SCL) for feature-level semantic consistency. Specifically, in ACL and SCL, the label information is used to constrain the attention and feature consistency of the target and the background, respectively, to avoid mutual interference. Extensive experiments have shown that our STCFormer outperforms many of the best-performing trackers on several popular benchmarks.

Introduction

Visual Object Tracking (VOT) has attracted increasing interest due to its broad applications in various fields, such as medical science (Bouget et al. 2017), unmanned aerial vehicle (Hao et al. 2018), and self-driving cars (Gao et al. 2020) which is a hot topic nowadays. One popular category of VOT is Single Object Tracking (SOT) and its goal can be described as follows: Given a target specified in the first frame, we need to estimate the states (including the location and the scale) of this target in subsequent frames (Soleimanitaleb and Keyvanrad 2022). Notwithstanding the booming of diverse tracking algorithms (Danelljan et al. 2018; Voigtlaender et al. 2019; Yan et al. 2021) in recent years, many challenges like appearance variation and occlusion still hinder us from accurate real-time tracking in complex real-world scenarios. Recapping current prevailing trackers, most of them cast this task as template-matching frame by frame. Specifically, it takes a target image as a template, and then compares it with other frames to find the location and the size of the target. The popular Siamese family (Bertinetto et al. 2016; Chen et al. 2020) is a typical representative of them. And the recently emerging Transformer-based trackers (Lin et al. 2022; Cui et al. 2022) realize template matching through full interaction between templates and search features.

Under the influence of this paradigm, a significant advancement in learning spatial representations has been made. But few have considered temporal information. Some handle this problem by adding an online updating mechanism into the inference stage (Zhang and Peng 2020; Yan et al. 2021), but this is trapped by template selection and update settings. Those who add complicated updating network (Bhat et al. 2019; Zhang et al. 2019) or memory network (Fu et al. 2021; Yang and Chan 2018) are plagued by massive computational complexity and bring in extra hyperparameters to tune. Besides, the common template fusion or similar operations also ignore the order of frames, which carries rich temporal contexts.

To alleviate these problems, we propose a **Space-Time Consistent Transformer Tracker (STCFormer)** to glean temporal information which is commonly overlooked by existing works. Specifically, we model the space-time relationship with multi-granularity cycle consistency constraints embedded in a sequential fusion framework, i.e. Label Consistency Loss (LCL), Attention Consistency Loss (ACL), and Semantic Consistency Loss (SCL). During training, we use a sequential fusion framework to construct a cycle that includes a template image and several chronological search images. We continuously update the template image based on previous prediction results during chronological frame tracking and use the final template for backward tracking with the first search frame again, forming a closed loop. To increase the robustness of our framework, we incorporate three consistency constraints to avoid over-dependence on

[*] These authors contributed equally.

[†] Corresponding Author

fixed templates. Intuitively, We expect the tracking results obtained from the same search image to be consistent, even if the template changes. For the first search image, we apply consistency constraints to its three levels of representation: label level with LCL to maximize the overlap of the two tracking results in the same search frame, patch level with ACL to ensure consistent attention weights of search patches based on template matching, and feature level with SCL to converge the semantics of filtered search features. Above strategies collaborate with one another to help the model learn more robust space-time features of the target. In particular, to strengthen ACL and SCL, we leverage label information to compute the consistency of patch attention and search features separating regions within the groundtruth box and the others, which prevents semantic confusion.

To sum up, our main contributions are:

- In the training process, we build a sequential fusion framework that forms a cycle by updating the template image based on previous prediction results during chronological frame tracking. With few hyperparameters, our method can be implanted into many existing Transformer-based trackers.
- To further mine the spatiotemporal information, we apply three consistency constraints of different granularities: Label Consistency Loss (LCL), Attention Consistency Loss (ACL), and Semantic Consistency Loss (SCL).
- We form a Space-Time Consistent Transformer Tracker (STCFormer) based on above strategies. Experiments on several widely-used benchmarks have proven its outstanding performance and verified the efficiency and efficacy of LCL, ACL and SCL.

Related Work

Transformer-based Trackers

According to the type of the framework, the SOT algorithms using Transformer can be classified as CNN-Transformer based trackers and fully-Transformer based trackers (Thangavel et al. 2023).

CNN-Transformer based trackers use hybrid architecture combining Convolutional Neural Network (CNN) and Transformer. And they adopt the Siamese-like pipeline as the majority of CNN-based trackers(Hu et al. 2023; Tan et al. 2021). Usually, the CNN is used for feature extraction and its outputs are fed into Transformer to catch the similarity between the template and the search region. Wang et al. (2021a) divides the encoder and decoder of the Transformer into two branches to build a Siamese-like tracker. Based on the DETR (Carion et al. 2020), Yan et al. (2021) develops an end-to-end framework to capture the global feature dependencies of both temporal and spatial information. Considering the neglect of the informative object-level information from those relative pixel positions, CSWinTT (Song et al. 2022) proposes multi-scale cyclic shifting window attention as a solution. Another observation about the correlation of query and key gives birth to AiATrack (Gao et al. 2022). It adds an inner attention to the Transformer structure.

Fully-Transformer based trackers is proposed to get rid of the reliance on CNN and they can catch global feature representations. DualTFR (Xie et al. 2021) is the trailblazer of this category. It wields a set of local attention blocks and a global attention block to seize both local and long-range dependencies. SwinTrack (Lin et al. 2022) extracts features by means of Swin Transformer (Liu et al. 2021) and reaches state-of-the-art performance at that time. All these are two-stream two-stage trackers because their backbone is a pre-trained Transformer while another Transformer is used for feature fusion and feature enhancement. Another type of fully-Transformer based tracker is the one-stream one-stage tracker. They utilize one Transformer to finish feature extraction and feature fusion, such as MixFormer (Cui et al. 2022). It comprises a set of Mixed Attention Modules for feature extraction and target information fusion but runs very slowly. Then Ye et al. (2022) devise OSTrack with ViT as the backbone. It comes up with an early candidate elimination strategy to delete background tokens by degrees. This benefits the accuracy and the speed at the same time.

Spatiotemporal Information Mining in Tracking

An influx of tracking algorithms has studied the spatial information over the past years. In a nutshell, they just consider tracking as a simple template-matching task in each frame. Nonetheless, a critical difference lies between image processing task (Carion et al. 2020) and video-based tracking (Lan et al. 2018) is the time-wise context. Thereupon some researchers probe into this issue. Methods based on optical flow (Senst, Eiselein, and Sikora 2012; Liu et al. 2020) achieve tracking an object by extracting the feature points and estimating their matching points in the next frame. Gao, Zhang, and Xu (2019) propose a spatiotemporal GCN to learn the structured representation of historical templates. TCTrack (Cao et al. 2022) explores adaptive temporal information at two levels, i.e., the feature extraction and the similarity maps refinement. An extra network to process several search images is also a common strategy. Zhang et al. (2019) equips SiamFC (Bertinetto et al. 2016) and DSiamRPN (Zhu et al. 2018) with a UpdateNet to learn the best template for the next prediction. Based on LSTM (Hochreiter and Schmidhuber 1997), MemTrack (Yang and Chan 2018) proposes a dynamic memory network to improve the model's adaptiveness to the target's appearance change. Though these do help in capturing some spatiotemporal relations between frames, still they are dependent on sophisticated structures and this inevitably leads to higher computational complexity. STARK (Yan et al. 2021) selects a simple and gradient-free mechanism of concatenating the initial template with a new one. But it is applied in the test stage only and it might cost almost double computational resources if applied in training. Those who incrementally update the model (Nam and Han 2015; Danelljan et al. 2016) are with the same shortage.

Consistency Loss

An interesting strategy of space-time information mining is cycle consistency loss. CycleGAN (Zhu et al. 2017) pioneered the method of cycle consistency to tackle unpaired image-to-image translation within two domains. Wu, Wang,

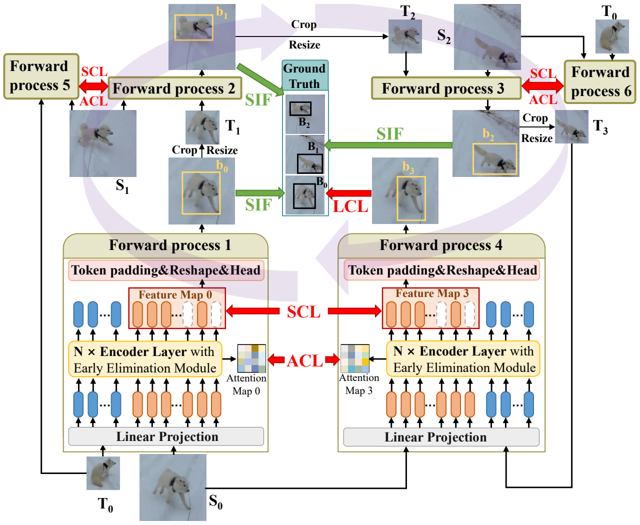

Figure 1: The overall pipeline of STCFormer. We conduct the customary forward process six times in one coherent forward circuit of STCFormer.

and Shao (2018) work on a similar idea for cross-modal retrieval. Wang, Jabri, and Efros (2019) and Wang et al. (2019) are the first to introduce this concept into object tracking. In spite of different implementations, both of them use this as a free supervisory signal to train unsupervised object trackers. And they both obtain impressive results. The successor (Yuan, Wang, and Chen 2020) simply expands the cycle consistency loss proposed by Wang et al. (2019) from DCFNet (Wang et al. 2017) into the self-supervised Siamese tracker. Dwibedi et al. (2019) propose a self-supervised algorithm to learn visual correspondence from unlabeled video. Analogously, Jabri, Owens, and Efros (2020) leverage cycle-consistency to align two similar videos in a self-supervised way. Ristea et al. (2023) introduce a similar idea into medical image processing and apply cycle consistency loss to both feature and computed tomography (CT) images. CCuantuMM (Bhatia et al. 2023) apply such idea into jointly matching multiple non-rigidly deformed 3D shapes. Above successful methods without supervisory signals motivate us to introduce such a powerful constraint into the supervised learning method.

Method

In this section, we present our Space-Time Consistent Transformer Tracker (STCFormer) in detail. We introduce the sequential fusion-based framework of STCFormer at first. Then we present our multi-granularity consistency constraints, i.e. Label Consistency Loss (LCL), Attention Consistency Loss (ACL) and Semantic Consistency Loss (SCL).

Framework of STCFormer

As shown in Fig. 1, the pipeline of our STCFormer is a coherent circuit rather than a common unidirectional structure. Define the routine of computing a predicted box with a template image and a search image as a *forward process* and denote it as Φ_θ (θ represents the model's parameters). Then one forward circuit of STCFormer is comprised of six forward processes. The input template image of each forward process (denoted as T_1, T_2, T_3) is cropped from the search image of its last forward process according to the last predicted box, except for the first forward process which crops a randomly sampled frame (denoted by F_{tem}) into T_0 as template image. And T_0 serves both forward process 5 and forward process 6. For the first three forward processes, the search images S_0, S_1, S_2 are respectively cropped from three frames F_0, F_1, F_2 that are sampled in chronological order. And the fourth forward process shares the same search image S_0 with the first forward process. The fifth forward process and the sixth forward process take S_1 and S_2 as input, respectively. The frames sampling can be described as:

$$F_{\text{tem}}, F_0, F_1, F_2 = \mathbb{S}\left(D, N_t, N_s\right), \tag{1}$$

where $\mathbb{S}(\cdot)$ refers to the sampling process; D is the training datasets; N_t, N_s represent the number of the template image and search images, respectively.

Aforementioned data processing procedure (leaving out data augmentations such as brightness jittering for brevity) can be written as:

$$\begin{cases} T_0 = \mathbb{C}_J\left(F_{\text{tem}}, B_{\text{tem}}, K_{\text{tem}}\right), \\ [S_0, S_1, S_2] = \mathbb{C}_J\left([F_0, F_1, F_2], [B_0, B_1, B_2], K_{\text{search}}\right), \end{cases} \tag{2}$$

where B represents the ground-truth bounding box of the corresponding frame, which is assigned by its subscript; $[F_0, F_1, F_2]$ represents the concatenation of three tensors; K_{tem} and K_{search} denote the size of the template image and the size of the search image, respectively. $\mathbb{C}_J(\cdot)$ is the jittering crop operation, which is used to avoid learning the bias that the target is always at the center of the search region.

On the basis of this circular framework, we establish Sequential Information Fusion (SIF) on results of the first three forward processes. Specifically, we impose a supervised constraint on predicted results of the first three forward processes and corresponding ground-truth labels. For each of the distance we utilize l_1 loss and the generalized IoU loss (Rezatofighi et al. 2019) for bounding box regression and the weighted focal loss (Law and Deng 2018) for classification. Then the SIF is defined as:

$$\mathcal{L}_{\text{SIF}} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathcal{L}_{\text{cls}}^{i} + \lambda_{\text{iou}}\,\mathcal{L}_{\text{iou}}^{i} + \lambda_{L_1}\mathcal{L}_{L_1}^{i}\right), \tag{3}$$

where n is the number of the search images and we set $n = 3$ in our experiments; $\lambda_{\text{iou}}, \lambda_{L_1}$ are the regularization terms and are set to 2, 5, respectively.

Based on this architecture, we design multi-granularity consistency loss LCL, SCL and ACL to narrow the gap between the outputs of those coupled forward processes with same search input but different template input, i.e., forward process 1 vs forward process 4, forward process 2 vs forward process 5, and forward process 3 vs forward process 6. Details of these space-time constraints will be depicted in following parts.

The differences between STCFormer and existing models can be summarized as:

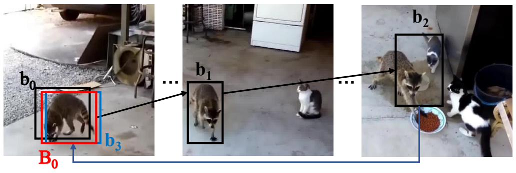

Figure 2: A defect of LCL.

- A classical space-wise tracker usually takes one template image and one search image as input. After one complete forward process, it outputs the state of the target in the search image. Distinct from it, the input of STCFormer is one template image and a series of search images. One coherent forward circuit of STCFormer contains six forward processes.
- Few space-time trackers have thought over the sequential information hidden in time flow, yet we construct a coherent forward circuit to process three search images in time order.
- Most space-time trackers directly fuse features of several template images or employing an elaborate network to create a new template based on them. Unlike these, we focus on the space-time consistency and design multi-granularity consistency loss (LCL, SCL and ACL) based on Sequential Information Fusion (SIF).

Label Consistency Loss (LCL)

The over-dependence of one fixed template prejudices the model's adaptiveness to the target's variation. Whereupon we update the template image stepwise. However, on account of most models' vulnerability to the template, arbitrarily changing the template image may cause opposite effects. To address this issue, we introduce Label Consistency Loss (LCL). The basic idea of it is to track forward along the time direction, followed by a backward tracking to form a closed loop. Its goal is to minimize the difference between the start and the end box coordinates. With one input template image T_0 and three search images sampled in chronological order S_0, S_1, S_2, this circular process can be formulated as below:

Firstly, we conduct the first forward process on T_0 and S_0 and obtain a predicted box b_0. Then we crop S_0 with b_0 as the label and resize it into a new template T_1. Afterward, we perform the second forward process with S_1 and T_1 as input. The third forward process is done in the same manner.

$$\begin{cases} b_i = \Phi_\theta\left(S_i, T_i\right), \\ T_{i+1} = \mathbb{C}_C\left(S_i, b_i, K_{\text{tem}}\right), \end{cases} \qquad i = 0, 1, 2, \tag{4}$$

where $\mathbb{C}_C(\cdot)$ means cropping an image with the given box region placed in the center of the cropped image.

Finally in the fourth forward process, T_3 (obtained by cropping S_2 and resizing) is combined with S_0 as the input and we can acquire a predicted box b_3 for S_0 in the end:

$$b_3 = \Phi_\theta\left(S_0, T_3\right). \tag{5}$$

Then our Label Consistency Loss (LCL) is defined as:

$$\begin{aligned} \mathcal{L}_{\text{cycle}}\left(B_0, b_3\right) = \mathcal{L}_{\text{cls}}\left(B_0, b_3\right) + \lambda_{\text{iou}}\mathcal{L}_{\text{iou}}\left(B_0, b_3\right) \\ + \lambda_{L_1}\mathcal{L}_{L_1}\left(B_0, b_3\right). \end{aligned} \tag{6}$$

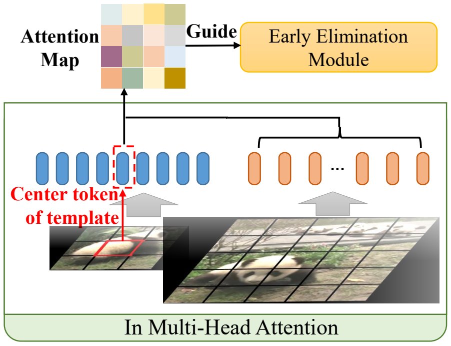

Figure 3: Details about the attention map of ACL.

Since LCL only works on the first label B_0 and the fourth predicted box b_3, a defect of it is that it may predict a precise box b_3 for S_0 while the interim box such as b_2 are not accurate, as Fig. 2 shows. Fortunately, we alleviate this problem with aforementioned SIF as it set constraints on each predicted box of the first three forward processes.

Attention Consistency Loss (ACL)

To further improve the robustness of tracking with dynamic template images, an intuitive idea is to train a model keeping its attention on where it should focus. In other words, even with different templates, the attention maps for the same search image should be consistent. That's exactly the key point of our Attention Consistency Loss (ACL). To avoid increasing the computational burden, we make full use of the calculation results of multi-head attention. In each encoder layer, multi-head attention calculates the correlation between input tokens. Following Ye et al. (2022), we take the similarity between the center token of the template image with all search tokens as the attention map. So the generation of attention map A_i for forward progress $(i+1)$ can be described as:

$$A_i = \begin{cases} \mathcal{A}\left(\mathbb{P}(T_i), \mathbb{P}(S_i), \Omega\right), & i = 0, 1, 2, \\ \mathcal{A}\left(\mathbb{P}(T_i), \mathbb{P}(S_0), \Omega\right), & i = 3, \\ \mathcal{A}\left(\mathbb{P}(T_0), \mathbb{P}(S_{i-3}), \Omega\right), & i = 4, 5, \end{cases} \tag{7}$$

where $\mathcal{A}(\cdot)$ is the multi-head attention operation; $\mathbb{P}(\cdot)$ represents the process involving three steps: (1) Split and flatten the image into sequences of patches; (2) Operate linear projection to transform patches into token embeddings; (3) Concatenate the template tokens and the search tokens as one input for cascaded Transformer encoder layers (Dosovitskiy et al. 2021). Ω is a generated mask for the template to retain the tokens of its center part.

To take full advantage of supervised signals that contains rich information, we divided search tokens into two parts: those that located within groundtruth box (including those that cross the boundary line of groundtruth box) are denoted as positive tokens while others are negative tokens. When we compute the distance between two attention maps A_i and A_j we perform normalization on them separately (denoted as $\Gamma(\cdot)$). Consequently, our ACL can be formulated as:

$$\mathcal{L}_{\text{attention}} = \frac{1}{3}\sum\left(\Gamma\left(\left|A_i - A_{i+3}\right|\right)\right), i = 0, 1, 2, \tag{8}$$

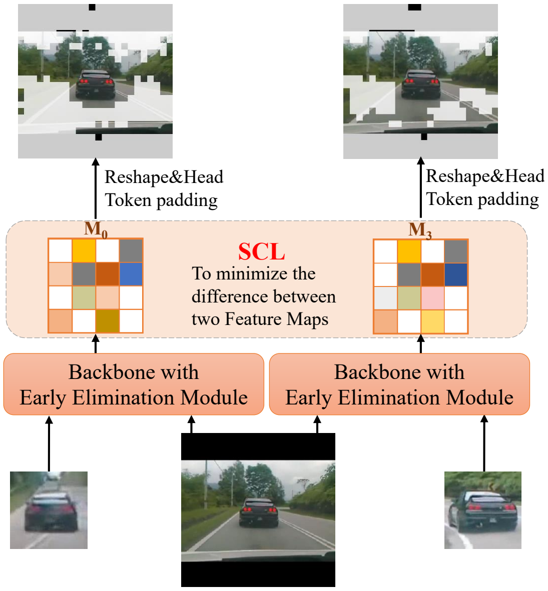
Reshape&Head
Token padding
Reshape&Head
Token padding
M0
M3
SCL
To minimize the difference between two Feature Maps
Backbone with Early Elimination Module
Backbone with Early Elimination Module

Figure 4: A schematic diagram of SCL.

where $|\cdot|$ denotes element-wise absolute operator.

Semantic Consistency Loss (SCL)

Since all the Transformer-based trackers rely on the data processing of divided input images into patches, we consider to grasp more fine-grained information which might have been omitted by them. Now with consistent label-wise information and patch-level attention weights, we expect the model to catch consistent semantics. More specifically, even if the template image changes (the target remains the same), the model is able to learn consistent semantic features for the same search image. This is reasonable as people are never restricted by one specific template during tracking. Instead, we capture the semantic information of the target to keep tracking regardless of appearance variation. This motivates us to learn robust semantic features for tracking.Since Ye et al. (2022) deletes several background tokens of the search image, the output tokens of the backbone are mostly belong to the target feature. Therefore we group these filtered tokens as a feature map and enforce the feature maps for the same search image as closely as possible while they are produced with different template images. Similar to ACL, we split the feature tokens into positive tokens and negative tokens and normalize them separately. In practice, we use MSE loss (denote as $E(\cdot)$) on those normalized feature pairs:

$$\mathcal{L}_{\text{semantic}} = \frac{1}{3}\sum(E\left(\Gamma(M_i), \Gamma(M_{i+3})\right)), i = 0, 1, 2, \quad (9)$$

In summary, the total loss of STCFormer is:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{SIF}} + \lambda_{\text{LCL}}\mathcal{L}_{\text{cycle}} + \lambda_{\text{ACL}}\mathcal{L}_{\text{attention}} + \lambda_{\text{SCL}}\mathcal{L}_{\text{semantic}}, \quad (10)$$

where the regularization terms λ_{LCL} , λ_{ACL} and λ_{SCL} are set to 1, 0.005, 0.1, respectively.

Experiments

This section introduces the implementation details at the beginning. Then we display the results of comparison with predominant algorithms. In the final part, we perform an ablation study to judge the contribution of each constraint and analysis our model from different perspectives.

Implementation

We implement STCFormer using Python 3.8 and PyTorch 1.9. It is trained on a server with 8 NVIDIA A100 GPUs. The inference speed is tested with only one NVIDIA RTX2080Ti GPU. Similar to OSTrack (Ye et al. 2022), the data augmentations such as horizontal flip and grayscale conversion are used in the training process.

Training. The batch size of each GPU is 28 and we train the model with AdamW optimizer (Loshchilov and Hutter 2017). The weight decay is 10^{-4}. The initial learning rate is 3×10^{-6} for the backbone and $3\times10^{-5}$for other parameters. We set total training epochs to 300 with 60k image pairs per epoch and the learning rate decreases by a factor of 10 after 240 epochs. The whole network is initialized with the training weights of OSTrack-384 (the search image is 384 × 384 pixels and the template image is 192 × 192 pixels) (Ye et al. 2022) and MAE (He et al. 2021).

Datasets. The model is trained with following datasets: COCO (Lin et al. 2014), LaSOT (Fan et al. 2018), GOT-10k (Huang, Zhao, and Huang 2018) and TrackingNet(Müller et al. 2018). And the benchmarks we used for test are GOT-10k (Huang, Zhao, and Huang 2018), LaSOT (Fan et al. 2018), LaSOT$_{\text{ext}}$ (Fan et al. 2020), TNL2K (Wang et al. 2021b) and UAV123 (Mueller, Smith, and Ghanem 2016).

Comparison with State-of-the-arts

We contrast STCFormer with 23 state-of-the-art approaches including SiamFC (Bertinetto et al. 2016), MDNet (Nam and Han 2015), ECO (Danelljan et al. 2016), SiamPRN++ (Li et al. 2018), DiMP (Bhat et al. 2019), ATOM (Danelljan et al. 2018), SiamR-CNN (Voigtlaender et al. 2019), LTMU (Dai et al. 2020), Ocean (Zhang and Peng 2020), KYS (Bhat et al. 2020), STMTrack (Fu et al. 2021), TrDiMP (Wang et al. 2021a), TransT (Chen et al. 2021), AutoMatch (Zhang et al. 2021), STARK (Yan et al. 2021), KeepTrack (Mayer et al. 2021), TransInMo (Guo et al. 2022), MixFormer (Cui et al. 2022), AiATrack (Gao et al. 2022), CIA (Pi et al. 2022), SwinTrack-B (Lin et al. 2022), GRM (Gao, Zhou, and Zhang 2023), and the baseline OSTrack-384 (Ye et al. 2022). Results are shown in Tab. 1.

LaSOT. Designed for long-term tracking, LaSOT is one of the most challenging large-scale benchmarks with 280 densely annotated testing videos. Compared with other powerful trackers, our STCFormer performs 0.4% higher than our baseline OSTrack-384 in AUC. And our scores in other two metrics (P$_{Norm}$ and P) also exceed other trackers.

LaSOT$_{\text{ext}}$. As an extended version of LaSOT, LaSOT$_{\text{ext}}$ contains 150 additional sequences of 15 object classes. Since it is released in recent two years, results on it are relatively fewer but we still set a new state-of-the-art on it with an AUC of 57.7% outperforming the baseline OSTrack-384 by

Method	Source	LaSOT			LaSOT$_{\text{ext}}$			TNL2K		GOT-10k			UAV123
		AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AUC	P	AO	SR$_{0.5}$	SR$_{0.75}$	AUC
SiamFC	ECCVW16	33.6	42.0	33.9	23.0	31.1	26.9	29.5	28.6	34.8	35.3	9.8	46.8
MDNet	CVPR16	39.7	46.0	37.3	27.9	34.9	31.8	31.0	32.2	29.9	30.3	9.9	-
ECO	ICCV17	32.4	33.8	30.1	22.0	25.2	24.0	32.6	31.7	31.6	30.9	11.1	53.7
SiamPRN++	CVPR19	49.6	56.9	49.1	34.0	41.6	39.6	41.3	41.2	51.7	61.6	32.5	61.3
DiMP	ICCV19	56.9	65.0	56.7	39.2	47.6	45.1	44.7	43.4	61.1	71.7	49.2	65.4
ATOM	CVPR19	51.5	57.6	-	-	-	-	40.1	39.2	-	-	-	65.0
SiamR-CNN	CVPR20	64.8	72.2	-	-	-	-	-	-	64.9	72.8	59.7	64.9
LTMU	CVPR20	57.2	-	57.2	41.4	49.9	47.3	-	-	-	-	-	-
Ocean	ECCV20	56.0	65.1	56.6	-	-	-	38.4	37.7	61.1	72.1	47.3	-
KYS	ECCV20	-	-	-	-	-	-	44.9	43.5	63.6	75.1	51.5	-
STMTrack	CVPR21	60.6	69.3	63.3	-	-	-	-	-	64.2	73.7	57.5	64.7
TrDiMP	CVPR21	63.9	-	61.4	-	-	-	-	-	67.1	77.7	58.3	67.5
TransT	CVPR21	64.9	73.8	69.0	-	-	-	-	-	67.1	76.8	60.9	69.1
AutoMatch	ICCV21	58.3	-	59.9	-	-	-	47.2	43.5	65.2	76.6	54.3	-
STARK	ICCV21	67.1	77.0	-	-	-	-	-	-	68.8	78.1	64.1	-
KeepTrack	ICCV21	67.1	77.2	70.2	48.2	-	-	-	-	-	-	-	69.7
TransInMo	CVPR22	65.7	76.0	70.7	-	-	-	52.0	52.7	-	-	-	69.0
MixFormer	CVPR22	70.1	79.9	76.3	-	-	-	-	-	71.2	80.0	67.8	70.4
AiATrack	ECCV22	69.0	79.4	73.8	-	-	-	-	-	69.6	80.0	63.2	70.6
CIA	ECCV22	67.6	-	71.5	-	-	-	50.9	-	67.9	79.0	60.3	68.9
SwinTrack-B	NeurIPS22	71.3	-	76.5	49.1	-	55.6	55.9	57.1	72.4	80.5	67.8	-
GRM	CVPR23	69.9	79.3	75.8	-	-	-	-	-	73.4	82.9	70.4	70.2
OSTrack-384	ECCV22	71.1	81.1	77.6	50.5	61.3	57.6	55.9	-	73.7	83.2	70.8	70.7
STCFormer	Ours	**71.5**	**81.5**	**78.0**	**52.0**	**63.0**	**59.6**	**57.7**	**59.0**	**74.3***	**84.2***	**72.6***	**70.8**

Table 1: Comparison with state-of-the-arts on five popular benchmarks: LaSOT, LaSOT$_{\text{ext}}$, TNL2K, GOT-10k and UAV123. The best results are shown in bold font. * means the figures are obtained following one-shot protocol.

1.5%. And our precision score and normalized precision score achieve improvements of 2% and 1.7%, respectively.

TNL2K. TNL2K is a new dataset for natural language guided tracking. To improve the generality of tracking evaluation it introduces several adversarial samples and thermal images which makes it more challenging. And we boost the performance in each metric. In AUC we surpass OSTrack-384 by 1.8% and outperform other powerful counterparts like SwinTrack-B and CIA by a substantial margin.

GOT-10K. GOT-10K is a widely-used large-scale benchmark that covers various common challenges in tracking. Its test set employs a one-shot tracking rule, which means that the trackers should only be trained on the GOT-10k training split, then object classes between train and test splits will not be overlapped. We follow this protocol and obtain improvements in AO, SR$_{0.5}$, SR$_{0.75}$ of 0.6%, 1.0%, 1.8%, respectively. We also perform favorably against the GRM (Gao, Zhou, and Zhang 2023), which is proposed in this year.

UAV123. UAV123 contains 123 videos, with an average of 915 frames per video. All frames of it are collected from low altitude aerial-views. On UAV123, we still have significant advantages over other trackers and show a slight increase of 0.1% in AUC.

Ablation Study and Analysis

Gains of each consistency loss. To judge the exact impact of LCL, ACL and SCL, we report the AUC scores of the models equipped with different constraints on three benchmarks. Results displayed in Tab. 2 prove that each constraint has made a contribution to the improvement of performance. More specifically, LCL benefits more on LaSOT and TNL2K, and ACL does well on LaSOT$_{\text{ext}}$. We guess there may be two reasons. (1) LCL sets constraints on both regression and classification, which deepen the space-time context learning. (2) feature-wise SCL pays attention to pixel-wise texture while patch-wise ACL work on higher-level information, which reveals global information of target features.

Utilization of supervised labels. In our three consistency losses, we fully utilize the label information from different perspectives. Actually, in the initial version, we conduct loss on the first predicted box b_0 and the backward predicted box b_3 in LCL, but in a bunch of experiments we found it unstable and the improvement is minimal. Then we turn to perform loss on the first ground-truth bounding box B_0 and the backward predicted box b_3, which leads to a noticeable growth on several benchmarks. Meanwhile, with SIF making B_0 and b_0 as close as possible, it is equivalent to closing the gap between b_0 and b_3. Based on SIF, we found the training process of new LCL becomes much more stable. Inspired by that, we introduce the label information into ACL and SCL to perform normalization.

Effect of weights on each loss. We also try different weights λ for each loss we design. The range of testing weights of each loss is approximately determined by the scale of their values. According to Tab. 3, LCL is unsusceptible to the hyperparameters (their training processes are almost the same), while the other two are not. Results show that 0.1 and 0.005 are best for SCL and ACL, respectively.

Speed and size. As Tab. 4 shows, in contrast to a selection of recent Transformer-based trackers, STCFormer achieves

OSTrack-384	LCL	ACL	SCL	LaSOT	LaSOT$_{\text{ext}}$	TNL2K
✓				71.1	50.5	55.9
✓	✓			71.4	51.0	57.6
✓		✓		71.2	51.7	57.5
✓			✓	71.3	50.9	57.5
✓	✓	✓	✓	71.5	52.0	57.7

Table 2: Quantitative comparison results of our tracker and its variants equipped with different loss functions.

Loss	LCL			ACL			SCL		
Weight	1	0.1	0.01	0.001	0.005	0.01	0.3	0.2	0.1
AUC	**52.0**	**52.0**	**52.0**	51.2	**51.7**	51	50.8	50.6	**50.9**

Table 3: Effect of different weights for LCL, ACL, SCL on LaSOT$_{\text{ext}}$. The best results are shown in bold font.

Tracker	Speed(fps)	MACs(G)	Params (M)
TrDiMP	26	-	-
TransT	50	-	-
STARK-ST101	32	18.5	42
SwinTrack-B-384	45	69.7	91
OSTrack-384	55.2	48.4	92
STCFormer (ours)	55.5	48.4	92

Table 4: Comparison of our inference speed and parameters with other representative Transformer-based trackers.

compelling computational and data efficiency. Our space-time consistency exploration almost cause no extra burden in model size. After code optimization, STCFormer even runs faster than the baseline OSTrack-384 in same environment.

Visualization. Fig. 5 exhibits some examples of our real-time tracking. The picture shows that STCFormer can handle many common challenges of tracking. It performs well under the circumstances of background clutters (row 1 and row 4). It can track accurately regardless of partial occlusion (with small or large occluded area) (row 2, row 7 and row 8). It is able to deal with deformation of the target (row 3) and catch small target (row 6). It can also distinguish the target from objects that look similar like it (row 5).

Conclusion

In this paper, we propose a novel Space-Time Consistent Transformer Tracker (STCFormer) based on a sequential information fusion framework. Multi-granularity consistency constraints i.e. Label Consistency Loss (LCL), Attention Consistency Loss (ACL) and Semantic Consistency Loss (SCL) are added to enhance the spatiotemporal consistency from label-level, patch-level and feature-level, respectively. Quantitative and qualitative analysis in experiments confirms the positive effect of our approach. In fact, our method is compatible with a wide array of contemporary trackers or even models for other visual tasks to further promote their performance by infusing more space-time information.

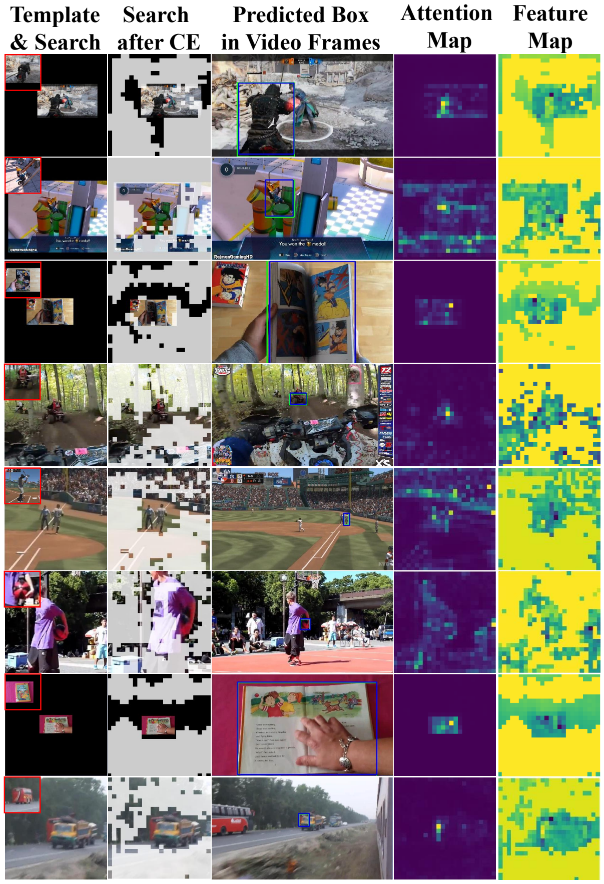

Figure 5: Visualization of the tracking process. The first column shows the search images (the big ones) and the template images (small ones in the upper left corner). The second column presents the search images after the early candidate elimination (CE) process of OSTrack. The third column shows the tracking results on corresponding frames. The green rectangles are groundtruth boxes and the purple rectangles are our predicted boxes. The fourth column shows attention map for corresponding search image and the fifth column is the filtered features after CE.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (91948303-1 & 62106278) and the Independent and open subject fund (grant no.202201-06) from State Key Laboratory of High Performance Computing.

References

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. S. 2016. Fully-convolutional Siamese networks for object tracking. In *ECCVW*, 850–865.

Bhat, G.; Danelljan, M.; Gool, L. V.; and Timofte, R. 2019. Learning discriminative model prediction for tracking. In *ICCV*, 6181–6190.

Bhat, G.; Danelljan, M.; Gool, L. V.; and Timofte, R. 2020. Know your surroundings: exploiting scene information for object tracking. In *ECCV*, 205–221.

Bhatia, H.; Tretschk, E.; Lähner, Z.; Benkner, M.; Möller, M.; Theobalt, C.; and Golyanik, V. 2023. CCuantuMM: Cycle-Consistent Quantum-Hybrid Matching of Multiple Shapes. In *CVPR*, 1296–1305.

Bouget, D.; Allan, M.; Stoyanov, D.; and Jannin, P. 2017. Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *Medical Image Analysis*, 35: 633–654.

Cao, Z.; Huang, Z.; Pan, L.; Zhang, S.; Liu, Z.; and Fu, C. 2022. TCTrack: temporal contexts for aerial tracking. In *CVPR*, 14778–14788.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*, 213–229.

Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; and Lu, H. 2021. Transformer tracking. In *CVPR*, 8122–8131.

Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; and Ji, R. 2020. Siamese box adaptive network for visual tracking. In *CVPR*, 6667–6676.

Cui, Y.; Cheng, J.; Wang, L.; and Wu, G. 2022. MixFormer: end-to-end tracking with iterative mixed attention. In *CVPR*, 13598–13608.

Dai, K.; Zhang, Y.; Wang, D.; Li, J.; Lu, H.; and Yang, X. 2020. High-performance long-term tracking with meta-updater. In *CVPR*, 6297–6306.

Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2016. ECO: efficient convolution operators for tracking. In *CVPR*, 6931–6939.

Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2018. ATOM: accurate tracking by overlap maximization. In *CVPR*, 4655–4664.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An image is worth 16x16 words: transformers for image recognition at scale. In *ICLR*.

Dwibedi, D.; Aytar, Y.; Tompson, J.; Sermanet, P.; and Zisserman, A. 2019. Temporal cycle-consistency learning. In *CVPR*, 1801–1810.

Fan, H.; Bai, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Harshit; Huang, M.; Liu, J.; Xu, Y.; Liao, C.; Yuan, L.; and Ling, H. 2020. LaSOT: a high-quality large-scale single object tracking benchmark. *IJCV*, 129: 439–461.

Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; and Ling, H. 2018. LaSOT: a high-quality benchmark for large-scale single object tracking. In *CVPR*, 5369–5378.

Fu, Z.; Liu, Q.; Fu, Z.; and Wang, Y. 2021. STMTrack: template-free visual tracking with space-time memory networks. In *CVPR*, 13769–13778.

Gao, J.; Zhang, T.; and Xu, C. 2019. Graph convolutional tracking. In *CVPR*, 4644–4654.

Gao, M.; Jin, L.; Jiang, Y.; and Guo, B. 2020. Manifold Siamese network: a novel visual tracking ConvNet for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21: 1612–1623.

Gao, S.; Zhou, C.; Ma, C.; Wang, X.; and Yuan, J. 2022. AiATrack: attention in attention for transformer visual tracking. In *ECCV*, 146–164.

Gao, S.; Zhou, C.; and Zhang, J. 2023. Generalized Relation Modeling for Transformer Tracking. *ArXiv*, abs/2303.16580.

Guo, M.; Zhang, Z.; Fan, H.; Jing, L.; Lyu, Y.; Li, B.; and Hu, W. 2022. Learning Target-aware Representation for Visual Tracking via Informative Interactions. In *IJCAI-22*, 927–934.

Hao, J.; Zhou, Y.; Zhang, G.; Lv, Q.; and Wu, Q. 2018. A review of target tracking algorithm based on UAV. In *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, 328–333.

He, K.; Chen, X.; Xie, S.; Li, Y.; Doll'ar, P.; and Girshick, R. B. 2021. Masked autoencoders are scalable vision learners. In *CVPR*, 15979–15988.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9: 1735–1780.

Hu, K.; Zhou, X.; Cao, M.; Wang, M.; Gao, G.; Yang, W.; and Tan, H. 2023. Progressive Perception Learning for Distribution Modulation in Siamese Tracking. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Huang, L.; Zhao, X.; and Huang, K. 2018. GOT-10k: a large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43: 1562–1577.

Jabri, A.; Owens, A.; and Efros, A. A. 2020. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*.

Lan, L.; Wang, X.; Zhang, S.; Tao, D.; Gao, W.; and Huang, T. S. 2018. Interacting Tracklets for Multi-Object Tracking. *IEEE Transactions on Image Processing*, 27(9): 4585–4597.

Law, H.; and Deng, J. 2018. CornerNet: detecting objects as paired keypoints. In *ECCV*, 765–781.

Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; and Yan, J. 2018. SiamRPN++: evolution of Siamese visual tracking with very deep networks. In *CVPR*, 4277–4286.

Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; and Ling, H. 2022. SwinTrack: A Simple and Strong Baseline for Transformer Tracking. In *NeurIPS*, volume 35, 16743–16754.

Lin, T.-Y.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: common objects in context. In *ECCV*, 740–755.

Liu, L.; Zhang, J.; He, R.; Liu, Y.; Wang, Y.; Tai, Y.; Luo, D.; Wang, C.; Li, J.; and Huang, F. 2020. Learning by analogy: reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, 6488–6497.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: hierarchical vision transformer using shifted windows. In *ICCV*, 9992–10002.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. In *ICLR*.

Mayer, C.; Danelljan, M.; Paudel, D. P.; and Gool, L. V. 2021. Learning target candidate association to keep track of what not to track. In *ICCV*, 13424–13434.

Mueller, M.; Smith, N. G.; and Ghanem, B. 2016. A Benchmark and Simulator for UAV Tracking. In *ECCV*, 445–461.

Müller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; and Ghanem, B. 2018. TrackingNet: a large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 310–327.

Nam, H.; and Han, B. 2015. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 4293–4302.

Pi, Z.; Wan, W.; Sun, C.; Gao, C.; Sang, N.; and Li, C. 2022. Hierarchical feature embedding for visual tracking. In *ECCV*, 428–445.

Rezatofighi, S. H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I. D.; and Savarese, S. 2019. Generalized intersection over union: a metric and a loss for bounding box regression. In *CVPR*, 658–666.

Ristea, N.-C.; Miron, A.-I.; Savencu, O.; Georgescu, M.-I.; Verga, N.; Khan, F. S.; and Ionescu, R. T. 2023. CyTran: a cycle-consistent transformer with multi-level consistency for non-contrast to contrast ct translation. *Neurocomputing*, 126211.

Senst, T.; Eiselein, V.; and Sikora, T. 2012. Robust local optical flow for feature tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 22: 1377–1387.

Soleimanitaleb, Z.; and Keyvanrad, M. A. 2022. Single object tracking: a survey of methods, datasets, and evaluation metrics. *arXiv preprint arXiv:2201.13066*.

Song, Z.; Yu, J.; Chen, Y.-P. P.; and Yang, W. 2022. Transformer tracking with cyclic shifting window attention. In *CVPR*, 8781–8790.

Tan, H.; Zhang, X.; Zhang, Z.; Lan, L.; Zhang, W.; and Luo, Z. 2021. Nocal-Siam: Refining Visual Features and Response With Advanced Non-Local Blocks for Real-Time Siamese Tracking. *IEEE Transactions on Image Processing*, 30: 2656–2668.

Thangavel, J.; Kokul, T.; Ramanan, A.; and Fernando, S. 2023. Transformers in single object tracking: an experimental survey. *arXiv preprint arXiv:2302.11867*.

Voigtlaender, P.; Luiten, J.; Torr, P. H. S.; and Leibe, B. 2019. Siam R-CNN: visual tracking by re-detection. In *CVPR*, 6577–6587.

Wang, N.; gang Zhou, W.; Wang, J.; and Li, H. 2021a. Transformer meets tracker: exploiting temporal context for robust visual tracking. In *CVPR*, 1571–1580.

Wang, N.; Song, Y.; Ma, C.; gang Zhou, W.; Liu, W.; and Li, H. 2019. Unsupervised deep tracking. In *CVPR*, 1308–1317.

Wang, Q.; Gao, J.; Xing, J.; Zhang, M.; and Hu, W. 2017. DCFNet: discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*.

Wang, X.; Jabri, A.; and Efros, A. A. 2019. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2561–2571.

Wang, X.; Shu, X.; Zhang, Z.; Jiang, B.; Wang, Y.; Tian, Y.; and Wu, F. 2021b. Towards more flexible and accurate object tracking with natural language: algorithms and benchmark. In *CVPR*, 13758–13768.

Wu, L.; Wang, Y.; and Shao, L. 2018. Cycle-consistent deep generative hashing for cross-modal retrieval. *IEEE Transactions on Image Processing*, 28: 1602–1612.

Xie, F.; Wang, C.; Wang, G.; Yang, W.; and Zeng, W. 2021. Learning tracking representations via dual-branch fully transformer networks. In *ICCVW*, 2688–2697.

Yan, B.; Peng, H.; Fu, J.; Wang, D.; and Lu, H. 2021. Learning spatio-temporal transformer for visual tracking. In *ICCV*, 10428–10437.

Yang, T.; and Chan, A. B. 2018. Learning dynamic memory networks for object tracking. In *ECCV*, 153–169.

Ye, B.; Chang, H.; Ma, B.; and Shan, S. 2022. Joint feature learning and relation modeling for tracking: a one-stream framework. In *ECCV*, 341–357.

Yuan, W.; Wang, M. Y.; and Chen, Q. 2020. Self-supervised object tracking with cycle-consistent Siamese networks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10351–10358.

Zhang, L.; Gonzalez-Garcia, A.; van de Weijer, J.; Danelljan, M.; and Khan, F. S. 2019. Learning the model update for Siamese trackers. In *ICCV*, 4009–4018.

Zhang, Z.; Liu, Y.; Wang, X.; Li, B.; and Hu, W. 2021. Learn to match: automatic matching network design for visual tracking. In *ICCV*, 13319–13328.

Zhang, Z.; and Peng, H. 2020. Ocean: object-aware anchor-free tracking. In *ECCV*, 771–787.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *ICCV*, 2242–2251.

Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; and Hu, W. 2018. Distractor-aware Siamese networks for visual object tracking. In *ECCV*, 103–119.