VideoTrack: Learning to Track Objects via Video Transformer

Fei Xie [†*], Lei Chu [‡], Jiahao Li [‡], Yan Lu [‡] and Chao Ma [†]
[†] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[‡] Microsoft Research Asia
jaffe031@sjtu.edu.cn, leichu@microsoft.com, Li.Jiahao@microsoft.com
yanlu@microsoft.com, chaoma@sjtu.edu.cn

Abstract

Existing Siamese tracking methods, which are built on pair-wise matching between two single frames, heavily rely on additional sophisticated mechanism to exploit temporal information among successive video frames, hindering them from efficiency and industrial deployments. In this work, we resort to sequence-level target matching that can encode temporal contexts into the spatial features through a neat feedforward video model. Specifically, we adapt the standard video transformer architecture to visual tracking by enabling spatiotemporal feature learning directly from frame-level patch sequences. To better adapt to the tracking task, we carefully blend the spatiotemporal information in the video clips through sequential multi-branch triplet blocks, which formulates a video transformer backbone. Our experimental study compares different model variants, such as tokenization strategies, hierarchical structures, and video attention schemes. Then, we propose a disentangled dual-template mechanism that decouples static and dynamic appearance clues over time, and reduces temporal redundancy in video frames. Extensive experiments show that our method, named as VideoTrack, achieves state-of-the-art results while running in real-time.

1. Introduction

Visual Object Tracking (VOT) is a fundamental problem in computer vision that aims to track an object of interest in a video given its bounding box in the first frame [53]. In recent years, mainstream approaches formulate visual tracking as a target matching problem, striking a good balance between performance and simplicity.

The philosophy of target matching is to find the object by looking for locations in the search area whose features have the largest similarity with those in the target template. However, target matching methods generally adopt per-frame object matching manner, where the rich temporal information in the video is largely overlooked. The representative methods are Siamese trackers [4, 20, 26, 27, 57]: pair-wise frames are fed into Siamese network to extract features and a matching network/operator is applied for target matching. Although recent pure transformer-based Siamese trackers [10, 16, 55, 61] unify feature extraction and matching into a single step by leveraging the Vision Transformer (ViT) [14, 31, 49], these Siamese trackers still follow a pair-wise matching philosophy that hinders their exploitation of temporal context.

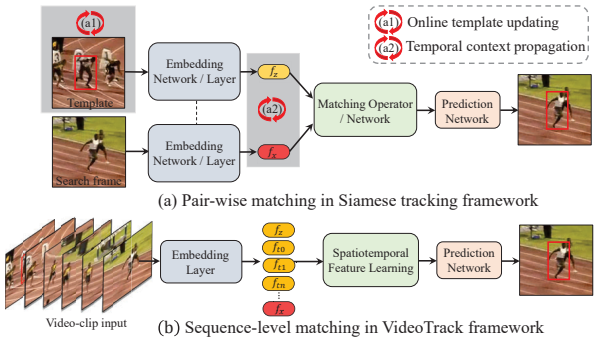


Figure 1. Comparing to the pair-wise matching pipeline in Siamese tracking shown in (a), which requires sophisticated mechanisms (a1)/(a2) to exploit temporal contexts, our neat video transformer tracking (VideoTrack) framework, as shown in (b), directly lifts the pair-wise feature matching into spatiotemporal domain.

To explore temporal information, some works have proposed sophisticated yet complex temporal modelling methods to improve the robustness of pair-wise Siamese trackers, where online template updating [60, 64] and temporal context propagating among frame-wise features [47] are two widely-adopted paradigms. Despite their great success, extra hand-crafted hyper-parameters and complex network modules are inevitably introduced to the Siamese pipeline, which have a negative impact on the efficiency and are not friendly to embedded devices. A natural question therefore arises: can we exploit the temporal context while still main-

*This work was done when Fei Xie was an intern at Microsoft Research Asia.

tain the tracking pipeline in a neat, end-to-end fashion?

To get rid of the customized updating strategies and redundant temporal modelling modules, we directly expand the pair-wise input frames into video-level (see Fig. 1), to capture rich temporal contexts. Specifically, we resort to video transformers [1] to learn spatiotemporal features, and establish the inter-frame temporal dependencies by simple feedforward modelling. Compared to the popular pair-wise Siamese matching [4, 27], our video transformer tracking pipeline (VideoTrack) lifts the 2D matching pipeline into the spatiotemporal domain, allowing the direct processing of video-level inputs. Moreover, we modify the video transformer architecture to better adapt it to tracking task, based on the following observations and prior knowledge:

Feature learning. A good feature representation is vital for the downstream vision tasks. Equipped with dedicated learning scheme in network layers, feature representations can be effectively enhanced from the shallow to deep level. Thus, we attempt to encode the temporal contexts at feature-level by utilizing a video transformer backbone. To ensure the generality and feasibility, we design our video transformer model with the following principles: 1) *Scalability*: Transformer layer is the basic building unit that can be stacked to construct the backbone network in different model scales. 2) *Compatibility*: To avoid expensive pre-training costs, the modified network should ideally be compatible with the model paramaters of the image-based vision backbone, *e.g*. ViT [14]. It not only can utilize the available pretraining weights, but also prevents the possible performance degeneration during fine-tuning.

Appearance vs. motion clue. Video could be viewed as a temporal evolution of a static appearance. Compared to the video recognition task which takes the complete frame as input, the input frames for most trackers are locally cropped from the online predicted target location, which weakens the motion clue in video-clips. Thus, we focus more on utilizing the appearance clues. To leverage the strong prior in video sequences, we explicitly divide them into three categories: initial frame containing strong appearance information, intermediate frames which contain the dynamic states of the target and search frame containing the target to be predicted. Thus, we formulate a *three-branch architecture* for the VideoTrack model.

Temporal redundancy. Consecutive video frames are highly redundant. It is vital to reduce the temporal redundancy as well as effectively modelling temporal contexts. Thus, we evaluate three basic temporal modelling approaches in terms of efficiency, *i.e*. joint space-time, temporal window and message token attention. With careful analysis, we propose a *disentangled dual-template mechanism* (see Sec. 3.4 for details) to integrate into the video backbone which decouples the redundant video information into the static & dynamic templates.

As shown in Fig. 2, we propose our VideoTrack framework on top of ViT [14], formulated by interleaving a series of building units, named as triplet-block. The triplet-block has three hierarchical attention layers that mix the information flow asymmetrically among three branches. Spatiotemporal guidance from the historical frames is passed to the current search frame, obtaining a compact feature representation for the final target prediction.

In summary, the main contributions are as follows:

- In contrast to existing Siamese tracking methods and their labor-intensive temporal modelling, we for the first time lift the 2D pair-wise matching to spatiotemporal domain, encoding temporal context at the feature-level via a neat feedforward video model, *i.e*. video vision transformer.
- We make the first attempt to adapt video transformer to visual tracking. A thorough ablation analysis of video transformer tracking is conducted, including tokenisation strategies, model variants and temporal modelling approaches. Comprehensive analysis may inspire followers to solve VOT task from the perspective of video-level modelling. Moreover, our tracker exhibits encouraging results on multiple VOT benchmarks.

2. Related Work

Visual tracking paradigm. The Siamese network [4, 8, 20, 26, 27, 57, 66] based tracking paradigms have drawn great attention recently, in which they formulate the tracking as per-frame target matching. Under the pair-wise matching framework, Siamese trackers are improved with the help of following techniques: powerful backbones [26, 66], elaborated prediction networks [20, 27, 57], attention mechanism [17, 48] and model fine-tuning [29, 46]. Recent pure transformer-based trackers [10, 16, 55, 61] leverage the vision transformer to unify the feature extraction and fusion, while still not consider how to effectively model the temporal dependency. Discriminative Correlation Filter (DCF) [5, 11–13, 23, 34, 62, 68] is another popular tracking paradigm, which can optimize the target model by solving least-squares based regression. Though DCF can easily utilize the temporal information by updating the model online, it suffers from the complex handcrafted optimization.

Temporal modelling in Siamese tracking. Two representative paradigms are introduced to enhance the temporal modelling in Siamese trackers: the first one is to update templates using online mechanism [60] or deep-learning based networks [17, 64]; the second one [47] is to propagate the target information from templates to search frame. Despite the improvements, both of them require extra hyper-parameters and redundant network modules to equip the original Siamese pipeline. In contrast to the sophisticated methods and tedious hyper-parameters mentioned above,

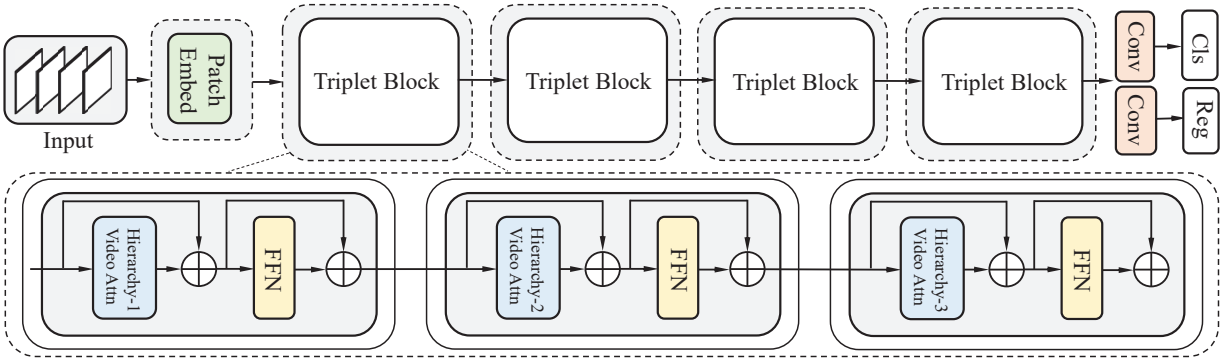


Figure 2. Overall architecture of video transformer model for tracking (VideoTrack). It is constructed by stacking multiple basic building units, named as triplet-block which consists of three hierarchical attention layers. Inside the each layer of triplet-block, video attention module mixes the multi-branch information flows among inputs asymmetrically. Then, the spatiotemporal learned features from search frame are fed to the prediction head for final target classification (Cls) and size regression (Reg). FFN denotes FeedForwad Network [45].

VideoTrack is the first to encode the temporal information via a simple feedforward video transformer backbone, which is novel and conceptually neat.

Video vision transformer. Video transformers have recently been introduced as powerful video recognition models, motivated by the impressive performance of transformers in language and vision [7, 9, 10, 14, 16, 18, 31, 45, 49, 51, 52, 58]. ViViT [1], Timesformer [3], VTN [37] and VideoSwin [32] are the pioneering works, which apply the pure-transformer based models for video recognition. The underlying reasons for their success lie in the characteristics of video: videos are sequential data while transformer attention can capture the global dependency among all the video segments. Considering visual tracking is highly sensitive to spatial/appearance information, rather than semantic/category, we modify the standard video transformer structure to exploit more static/dynamic appearance clues for tracking.

Temporal modelling in video understanding. Temporal context modelling is the key issue in video understanding task. 3D convolutional block [6, 39, 43, 44, 56] is the widely adopted technique, which expands 2D CNN into temporal domain. Then, non-local network [50] applies self-attention to capture long-range spatiotemporal dependencies on top of 2D CNN. Recently, video transformers [1, 3] use self-attention as the exclusive building block to capture spatiotemporal context. In instance-level video understanding, temporal shift [59, 63] and message token [25] mechanisms are equipped into video transformer to enhance the temporal modelling as well as reducing the computation cost. In this work, we empirically evaluate different temporal modelling methods and develop a disentangled dual-template scheme for VideoTrack model.

3. Proposed Method

In this section, we briefly introduce the vision transformer based Siamese tracking architecture, denoted as ViTtrack. Then, we expand it to temporal domain for processing video-level input. Finally, we present necessary modifications to the video transformer and propose a disentangled dual-template mechanism to better adapt it to tracking task.

3.1. Revisiting Vision Transformer Tracking

The attention mechanism of transformer [45] has been applied in VOT for feature extraction and fusion [7, 10, 16, 58, 61]. ViTtrack adapts the transformer architecture of [14, 49] to process 2D template-search image pair with minimal changes. In particular, ViTtrack extracts $\{N_x, N_z\}$ non-overlapping image patches $\{x_i \in \mathbb{R}^{h\times w}, z_i \in \mathbb{R}^{h\times w}\}$, for search image and template image, respectively. The sequence of tokens input to the following vision transformer layer is:

$$f_{zx} = [f_z, f_x] = [\mathbf{E}x_1, \ldots, \mathbf{E}x_{N_x}, \mathbf{E}z_1, \ldots, \mathbf{E}z_{N_z}], \quad (1)$$

where the projection by $\mathbf{E}$ is performed by a 2D convolution, and we omit the positional encoding here for simplicity. The tokens are then passed through vision transformer backbone consisting of a sequence of L transformer layers. Each layer ℓ comprises of Multi-head Self-Attention (MSA) [45], Layer Normalisation (LN) [2], and a Multi-Layer perceptron (MLP) [14] as follows:

$$\begin{aligned} y^{\ell}_{zx} &= \text{MSA}(\text{LN}(f^{\ell}_{zx})) + f^{\ell}_{zx}, \\ f^{\ell+1}_{zx} &= \text{MLP}(\text{LN}(y^{\ell}_{zx})) + y^{\ell}_{zx}, \end{aligned} \quad (2)$$

where the MLP consists of two linear projections separated by a GELU non-linearity [22]. The token-dimensionality in vanilla ViT [14] remains fixed throughout all layers. Recently, other improved ViTs [31, 49, 52] adopt multi-scale structure which gradually expand the channel dimension and reduce the spatial size. We omit the multi-scale structure in Eq. 2 for simplicity. Then, the template and search image features $\{f_z, f_x\}$ are jointly extracted and fused through multiple attention layers. In the final stage,

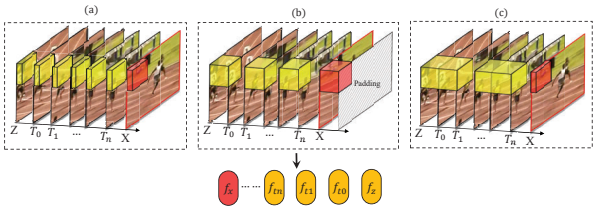


Figure 3. Three different embedding methods. (a) separated embedding. (b) tubelet embedding. (c) combination of separated (search frame) and tubelet embedding (templates).

a prediction network is used to decode the fused search image feature f_x^L to locate and estimate the size of target:

$$y_{reg} = \Phi_{reg}(f_x^L), \quad y_{cls} = \Phi_{cls}(f_x^L), \tag{3}$$

where $\{y_{reg}, y_{cls}\}$ denote the target location and shape estimation results. $\{\Phi_{cls}, \Phi_{reg}\}$ are classification and regression head. In next section, we present how to expand the pair-wise ViTtrack architecture into video-level and specialized adaptations to VOT task.

3.2. Expanding Siamese Matching to Video Level

We first present the sampling strategies to construct video-clips briefly. Then, to process video-level inputs, we discuss two pillars in video transformer model: token embedding and video attention scheme.

Video-clip sampling strategy. We construct the video-clip $\{z, x, t_0, t_1, ..., t_T\}$ with the strong prior on VOT task: the search frame x and first template z with ground truth are always available. To get the fixed number (T) of intermediate frames t, a common strategy is uniform sampling where each frame is distributed uniformly on the temporal extent [3]. An alternative is to maintain a memory queue for intermediate frames $\{t_0, t_1, ..., t_T\}$, which is updated with fixed temporal interval. These two strategies do not have key impacts on tracking performance when videos are not long. In this work, we select the memory queue strategy for its flexibility to handle videos in different length.

Video-level token embedding. We first consider two embedding methods: *separated frame embedding* [3] and *tubelet frame embedding* [1] for mapping a video-clip. In Fig. 3 (a), separated frame embedding is a straightforward method of tokenising the video frames which embeds each 2D frame independently [3]. An alternate method, tubelet frame embedding, is to extract non-overlapping, spatio-temporal tubes and embed them into tokens through 3D convolutional layers [1]. As shown in Fig. 3 (b), this method fuses spatio-temporal information during tokenisation, in contrast to separated frame embedding without inter-frame temporal fusion. As the feature embeddings in shallow layers may not formulate a rich representation [28, 54] and the prediction is only conducted on the search frame, early interactions may contaminate the search frame. Thus, we also perform the combination of two embedding methods, where tubelet frame embedding is only for templates and separated embedding for the search frame (see Fig. 3 (c)). More discussions can be found in Sec. 4.2.

Video transformer attention. In contrast to the classical 2D transformer in Siamese tracking [7, 10, 16, 58], video transformer is required to process the feature tokens which have longer temporal extents. Moreover, temporal modelling scheme of video transformer layer needs to be adapted to VOT task and we present it in Sec. 3.3.

3.3. Adapting Video Transformer to Tracking

In this section, we further adapt video transformer model to visual tracking in the following aspects: network structure, video attention and temporal modelling.

Stacked hierarchical structure. As shown in Fig. 2, we propose a basic VideoTrack architecture which has multiple stacked triplet-blocks. Triplet-block, as the basic construction unit, is composed of three standard transformer layers in a sequence. Inside the triplet-block, hierarchical attention computation for the video inputs is performed:

$$\begin{aligned} v^{l+1} &= \text{MSTB}_l^i(v^l), \quad v \in \{z, x, t\}, \\ v^{l+2} &= \text{MSTB}_{l+1}^i(v^{l+1}), \quad v \in \{z, x, t\}, \\ v^{l+3} &= \text{MSTB}_{l+2}^i(v^{l+2}), \quad v \in \{z, x, t\}, \end{aligned} \tag{4}$$

where MSTB_l^i denotes the l^{th} layer in i^{th} Multi-head Self-attention Triplet-Block and v denotes video-clip tokens. For clarity in this work, we let the $\{z, x, t\}$ denote the corresponding feature tokens in each layer instead of $\{f_z, f_x, f_t\}$ and t indicates all intermediate templates whose the quantity is T. Thus, VideoTrack can be scaled up to large model capacity by simply stacking more triplet-blocks.

Integrated into original ViT model. As we keep VideoTrack be compatible with ViT [14] in terms of model parameter, so VideoTrack can be built on ViT-Base [14] (L=12, N_H=12, d=768) seamlessly, where L is the number of transformer layers, each with an attention module of N_H heads and hidden dimension d. It avoids the specialized initialization on the incompatible parameters, thus prevents the possible performance degeneration during fine-tuning.

Multi-branch asymmetric attention. We explicitly divide the video inputs into three categories: the search image x which needs to predict the target online, the first template z containing the strong appearance clue of our target and intermediate templates $\{t_0, t_1, ..., t_T\}$ which store the dynamic states of the tracked target. In Fig. 4 and Fig. 5, the inputs $\{z, x, t_0, t_1, ..., t_T\}$ formulate three branches inside the block, and the video-level attention matching can be flexibly arranged to mix the information asymmetrically. In this work, we adopt vanilla self/cross attention scheme [45] without tricks (see Sec. 4.2 for more discussions).

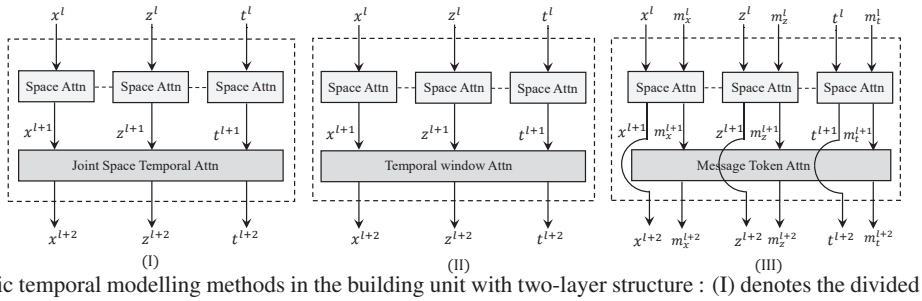


Figure 4. Three basic temporal modelling methods in the building unit with two-layer structure : (I) denotes the divided space time pattern; (II) denotes the temporal window pattern; (III) denotes the pattern with message token for temporal modelling. The number of total layers in all building units is 12. The dash line indicates the weight-sharing of convolutional layers. We omit the MLP and residual connection for simplicity. t^l denotes all the tokens from intermediate templates $\{t_0^l, t_1^l, ..., t_T^l\}$ in l^{th} layer unless specified. Best viewed with zooming in.

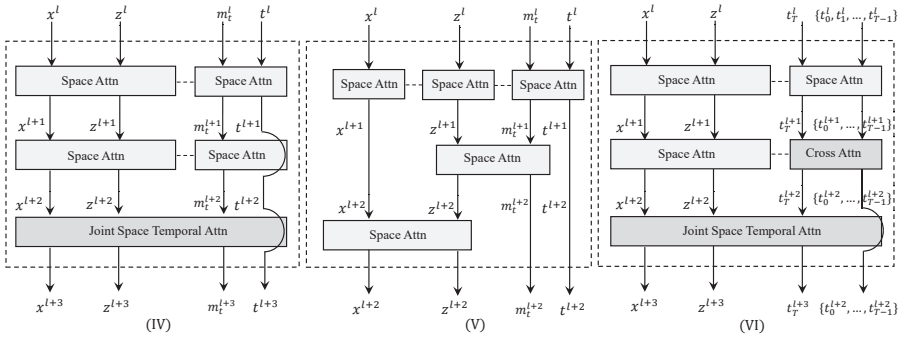


Figure 5. Three different building units with three-layer hierarchical structure (triplet-block): (IV) denotes the enhanced message token pattern; (V) denotes the message token attention with separated layer pattern; (VI) denotes the disentangled dual-template video attention pattern. we still keep the number of total layers as 12. Other annotations can be referred to Fig. 4. More conclusions on the pattern design can be found in appendix. Best viewed with zooming in.

Temporal context modelling. We present three basic methods to model the temporal information in video attention: joint space-time, temporal window and message token attention. A direct way to modelling the temporal context is to compute the dense attention among all spatiotemporal tokens across space-time:

$$\mathbf{y}_{st} = \sum_{s't'} \mathbf{v}_{s't'} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{s't'} \rangle}{\sum_{\bar{s}\bar{t}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{\bar{s}\bar{t}} \rangle}, \tag{5}$$

where query-key-value vectors $\mathbf{q}_{st}, \mathbf{k}_{st}, \mathbf{v}_{st}$ are the linear projections of corresponding feature token from each space-time location st in video frames. For clarity, we neglect the LN and replace the MSA with single-head in Eq. 5. Obviously, quadratic complexity in both space and time, *i.e.*, $O(S^2T^2)$, hinders it from capturing longer temporal contexts, as the temporal redundancy rises dramatically. An alternative is to partition the video input into windows along temporal dimension: 3D windows are arranged to partition the video input in a non-overlapping manner, then the attention computes within it (refer to [32] for technical details).

To further reduce the computation cost, message token communication [25,59] utilizes pre-defined fixed tokens m^t to summarize the per-frame context information and use it for temporal propagation:

$$\begin{aligned} &a^{l+1}, m_a^{l+1} = \text{MSA}_l([a^l, m_a^l]), \quad a \in \{z, x, t\}, \\ &m_z^{l+2}, m_x^{l+2}, m_t^{l+2} = \text{MSA}_{l+1}([m_z^{l+1}, m_x^{l+1}, m_t^{l+1}]), \end{aligned} \tag{6}$$

where $[..., ...]$ denotes concatenation among tokens. m_a is expanded from the learned message tokens m for each frame-wise tokens a. We empirically evaluate mentioned methods and find that they either cannot convey thorough appearance clues for tracking or not keep affordable computation cost for long temporal extents (see Sec. 4.2).

3.4. Disentangled Dual-Template Mechanism

By analyzing above three basic methods in terms of temporal modelling efficiency, we propose a disentangled dual-template mechanism to decouple the static and dynamic state of the appearance information across time. The main idea is to explicitly leverage the strong static appearance information from the first template and the dynamic factors of the intermediate templates through the efficient temporal modelling. It reduces the computation & temporal redundancy in joint space-time attention by only performing the cross attention among intermediate templates, while propagating the appearance information more thoroughly than the message token communication. As shown in Fig. 5 (VI), our mechanism is applied to the triplet-

case	Pattern I	Pattern VI
Separate	**71.3**	**70.6**
Tubelet for t	66.4	65.8
Tubelet	65.3	63.1

Table 1. Ablations on input encoding. The performance is AO in GOT-10k [24].

case	Pattern I	Pattern VI
None	70.1	69.5
Space-only	70.5	69.8
Space-Time	**71.3**	**70.7**

Table 2. Ablations on positional embedding. The performance is AO in GOT-10k [24].

case	Pattern I	Flops	Param.
Vanilla ViT [14] Attn	71.3	67.7 G	**85.4** M
VideoSwin [32] Attn	69.2	**49.6** G	90.9 M
Trajectory [38] Attn	**71.5**	70.5 G	**85.4** M

Table 3. Ablations on video attention design. The performance is AO in GOT-10k [24].

block. The spatiotemporal matching among the video tokens $\{z^l, x^l, t_0^l, t_1^l, ..., t_T^l\}$ in l^{th} triplet-block is computed by three-level asymmetric attention. The intermediate templates $\{t_0^l, t_1^l, ..., t_T^l\}$ are first encoded by the spatial layer respectively, then the nearest intermediate template t_T^{l+1} performs cross attention to aggregate the dynamic information from other templates $\{t_0^{l+1}, t_1^{l+1}, ..., t_{T-1}^{l+1}\}$. In the third layer, the generated dynamic template t_T^{l+2} is added to the matching between the z^{l+2} and x^{l+2}:

$$\begin{aligned} t_i^{l+1} &= \text{MSTB}_l^i(t_i^l), \quad i \in \{1, 2, ..., T\}, \\ t_T^{l+2} &= \text{MCTB}_{l+1}^i(t_T^{l+1}, [t_0^{l+1}, t_1^{l+1}, ..., t_{T-1}^{l+1}]), \\ z^{l+3}, x^{l+3}, t_T^{l+3} &= \text{MSTB}_{l+2}^i([z^{l+2}, x^{l+2}, t_T^{l+2}]), \end{aligned} \tag{7}$$

where $\text{MCTB}_l^i(A, B)$ indicates the l^{th} layer in i^{th} Triplet-Block that conducts Multi-head Cross-attention (A performs as query and B as key/value). $[..., ...]$ denotes concatenation among tokens. Here, we omit the matching between the z and x in first two layers for clarity.

4. Experiments

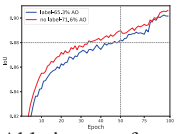
This section firstly describes the implement details and experimental settings. Then, we conduct a comprehensive analysis on the VideoTrack framework and compare it to state-of-the-art (sota) trackers on four VOT benchmarks.

4.1. Implementation Details

Network architecture and training. Our VideoTrack architecture is on top of ViT-Base [14] (L=12, N_H=12, d=768). We use the COCO [30], LaSOT [15], GOT-10k [24] and TrackingNet [36] as training datasets. Considering the memory restriction, we set the T as 2. Thus, each batch has 4 frames, resulting in total batch size of 128 for 8 Tesla V100 GPUs. The template and search image size

case	Pattern I	Pattern VI
None	62.1	60.5
ImageNet-1k [14]	70.4	70.2
ImageNet-21k [14]	71.2	70.8
VideoMAE [41]	64.1	63.5
MAE [21]	**73.3**	**72.6**

Table 4. Ablations on pretrain. MAE pre-training is more effective. The performance is AO in GOT-10k [24].



case	Pattern I
Label	65.3
No label	**71.6**

Figure 6. Ablations on foreground label. It shows the IoU curve during training and their best performance in GOT-10k [24].

are cropped as 128 and 256, respectively. The total training epochs are set to 300 with 60k sequences per epoch. More details can be found in appendix.

Online inference. During inference, our VideoTrack model is independent of the number of frames. In this work, for the current search frame F_t, we select fixed number ($T = 4$) of frames from historical frames (*i.e.* frame $F_{t-N\times T}$ to frame F_{t-1}) as intermediate templates. Each intermediate template is sampled from the nearest frame with the fixed time interval N, where we set to 30 frames here. When current time index t is over or less than the capacity of memory queue, we drop the farthest intermediate template or duplicate the first template at once.

4.2. Ablation study

In this section, we first evaluate the design choices regarding with processing video-level inputs. Then we explore the temporal modelling ability of different model variants. In the final, we study two additional commonly concerned aspects: pretraining and usage of foreground label. More conclusions can be found in appendix.

Input encoding. As the input encoding is vital to process sequence-level frames, we first consider the two widely adopted encoding methods in action recognition and their combinations. In Tab. 1, we study the effect of separated and tubelet frame embedding applied to the templates and search image. Separated embedding in two model patterns surpasses the tubelet embedding by a large margin (71.3% vs. 65.3% in pattern I and 70.6% vs. 63.1% in pattern VI). It is obvious that early feature interactions between search image and template images make the model difficult to predict the tracked target. By comparing the separated embedding to tubelet embedding only for templates (71.3% vs. 66.4% in pattern I), we also find similar phenomenon that early feature fusion between templates deteriorates the tracking performance. Thus, we suggest the separated embedding for inter-/intra-frame patches to generate spatiotemporal tokens.

T	Pattern I	Pattern II	Pattern III	Pattern IV	Pattern V	Pattern VI
1	73.1	67.5	62.6	**72.6**	**71.5**	72.1
2	**73.4**↑	68.2 ↑	**63.2**↑	71.5↓	71.2 ↓	72.3 ↑
3	72.2↓	**68.3**↑	62.8 ↓	71.0 ↓	70.8 ↓	72.6 ↑
4	70.3↓	67.1 ↓	62.1↓	70.6↓	70.2 ↓	**72.7** ↑

Table 5. Ablations on intermediate templates in four model structures. The performance is AO in GOT-10k [24] and the arrow (↑/↓) shows the trend when frame number increases.

T	1	2	3	4	5	8	10
Pattern I	69.4	69.7	**69.8**	69.3	68.7	68.1	67.7
Flops(G)	32.7	38.3	**43.8**	49.3	54.7	71.2	82.1
FPS	102.3	92.1	**89.2**	65.2	60.1	52.3	42.7
Pattern VI	69.2	69.3	69.8	**70.1**	69.9	68.8	68.6
Flops(G)	32.8	35.1	37.4	**39.7**	42.1	49.0	53.6
FPS	101.1	97.8	85.9	**81.4**	68.8	65.2	55.3

Table 6. Ablations on the length of input intermediate template during inference. The performance is AO in LaSOT [15].

Position embedding. As the position embedding is vital for the model to discriminate the video frames in both spatial and temporal domain, we conduct experiments with a few model variants that use: (1) no Positional Embedding (PE), (2) space-only PE, and (3) space-time PE (refer to [3]). Based on these results in Tab. 2, we observe that the variant of our model that uses space-time PE produces the best AO performance (71.3% and 70.6%) on both two model patterns. Interestingly, we also observe that the gap between space-time PE and space-only PE is larger than that of between space-only and not using PE (0.8% vs. 0.4% and 0.9% vs. 0.3%). This makes sense as the video-level input processing demands complex temporal reasoning.

Attention scheme. It is non-trivial to design attention scheme while the actual gain of specialized design remains open. Under the same setting, we evaluate three attention variants in Tab. 3: vanilla attention [14], 3D shifted window attention in VideoSwin [32] and trajectory attention [38]. We find that VideoSwin (69.2%) and trajectory attention (71.5%) do not have obvious advantages to the vanilla attention (71.3%) in terms of performance and model cost. This is because the VideoSwin and trajectory attention are the approximation of vanilla attention and their improvements are mainly designed for the high-level video understanding tasks, *e.g.*, action recognition. So we adopt the vanilla attention in this work for simplicity and efficiency.

Temporal modelling. As shown in Fig. 4, we first evaluate three fundamental temporal modelling using the same two-layer hierarchical transformer structure (pattern I, II and III). The results are shown in Tab. 5. The pattern I achieves the best 73.1% AO performance indicating that space-time attention has stronger temporal modelling ability than the time window (68.3%) and message token attention (63.2%). However, the performance of three patterns degrades as the frame number increases, which reveals their weakness at capturing long-range temporal contexts. Then, we add one more transformer layer to enhance the temporal modelling (pattern IV, V and VI), which are shown in Fig. 5. Both the three-layer patterns IV (72.6%) and pattern V (71.5%), improve the performance by large margin comparing to the pattern II (62.6%). However, the performance degeneration when the frame number increases still exits. As our proposed disentangled dual-template mechanism (pattern VI) reduces the temporal redundancy in intermediate templates by cross attention, its performance has a rising tendency facing the longer video-clip (72.1% to 72.7%). Notice that the videos in GOT-10k [24] is not long, it may explain why the improvements is relatively small. We further validate it in long video benchmarks.

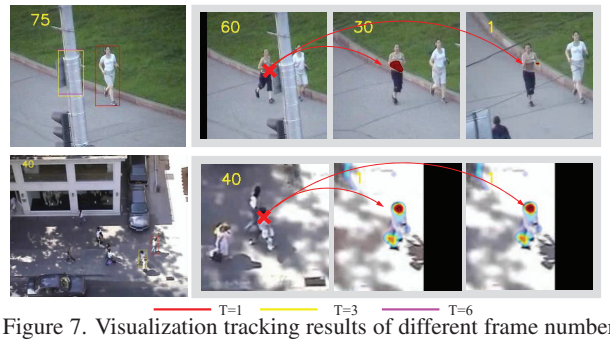


Figure 7. Visualization tracking results of different frame number in video input and their heat maps ($T = 3$) of the cross attention in the intermediate layer of triplet-block. We can see that our dual template mechanism can adaptively exploit the temporal information. Best viewed with zooming in.

Varying the number of input frames. The scalability of VideoTrack allows it to operate on longer videos. In Tab. 6, we further compare the pattern I and pattern VI in terms of efficiency on the long video benchmark LaSOT [15]. The best performance of pattern VI surpasses pattern I by 0.3% with more input frames ($T = 4$ vs. $T = 3$). It is worth noting that pattern VI achieves 9.4% computation Flops reduction and raises 16.2 FPS inference speed, comparing to the model I. It validates that our proposed disentangled dual-template mechanism can reduce the temporal redundancy and effectively convey appearance information. Observing that both two patterns cannot benefit from over long video frames, we infer that over-expanding the temporal extents does not provide more meaningful appearance clues but harm the model efficiency. It is consistent with our motivation to propose pattern VI. Moreover, the performance drop also comes from the online tracking error accumulation which deteriorates the quality of templates.

Pretrain. We empirically evaluate the pretrained weight from ImageNet-1K [40], ImageNet-22k [42] and recent proposed MAE [21, 41] method. The results of Tab. 4 show that MAE [21] pretraining outperforms the other pretraining methods (73.3% vs.70.4%). However, we also observe that VideoMAE [41] pretraining does not perform well which only achieves 64.1% AO, comparing to the 70.4% AO from

	SiamRPN++ [26]	ATOM [12]	DiMP [5]	Tr Siam [47]	TransT [7]	Stark st101 [58]	Mixformer 1k large [10]	SBT large [16]	OStrack 256 [61]	**VideoTrack**
$AO\uparrow$	51.8	55.6	61.1	66.0	67.1	68.8	67.9	**70.4**	**71.0**	**72.9**
$SR_{50}\uparrow$	61.6	63.4	71.7	76.6	76.8	78.1	77.3	**80.8**	**80.4**	**81.9**
$SR_{75}\uparrow$	32.5	40.2	49.2	57.1	60.9	64.1	63.9	**64.7**	**68.2**	**69.8**

Table 7. Comparison on the GOT-10k [24] test set.

	SiamRPN++ [26]	ATOM [12]	DiMP [5]	Tr Siam [55]	TransT [47]	DTT [7]	Stark s50 [62]	Mixformer 1k [58]	SBT large [10]	OStrack 256 [16]	**VideoTrack** [61]
AUC↑	49.6	51.5	56.9	62.4	64.9	60.1	65.8	**67.9**	66.7	**69.1**	**70.2**
Prec↑	49.1	50.5	56.7	60.0	69.0	-	69.7	**73.9**	71.1	**75.2**	**76.4**

Table 8. Comparison on the LaSOT [15] test set.

	ECO [11]	SiamFC [4]	SiamFC++ [57]	PrDiMP [13]	D3S [33]	AutoMatch [65]	TransT [7]	Stark st50 [58]	Mixformer 1k [10]	Ostrack 256 [61]	**VideoTrack**
AUC↑	55.4	57.1	75.4	75.8	72.8	76.0	81.4	81.3	**82.6**	**83.1**	**83.8**
Norm.Prec↑	61.8	66.3	80.0	81.6	76.8	82.4	86.7	86.1	**87.7**	**87.8**	**88.7**
Prec↑	49.2	53.3	70.5	70.4	66.4	72.5	80.3	-	**81.2**	**82.0**	**83.1**

Table 9. Comparison on the TrackingNet [36] test set.

	ATOM [12]	Ocean [67]	AutoMatch [65]	SiamGAT [19]	Tr DiMP [47]	TransT [7]	Stark st50 [58]	Mixformer 1k [10]	OStrack 256 [61]	**VideoTrack**
AUC↑	61.7	62.1	64.4	64.6	67.0	68.1	**69.2**	**68.7**	68.3	**69.7**
Prec↑	82.7	82.3	83.8	84.3	87.6	87.6	**88.2**	**89.5**	-	**89.9**

Table 10. Comparison on the UAV123 [35] test set.

image classification pretraining. This is explained by the fact that video datasets have much temporal redundancy and image-based pretraining can help VideoTrack to capture more appearance clues than motion clues.

Foreground label map. We study the necessity to use foreground label maps in the backbone network as shown in Fig. 6. Foreground label maps are added to the templates in video inputs, which is similar to the STMtrack [17]. The setting with label has a faster convergence, but lower tracking performance (65.3% vs. 71.6%). Besides the overfitting caused by rich template clues, encoding the temporal contexts at the feature-level makes the online temporal error accumulation contaminate the feature representations more easily. Thus, we suggest not using explicit foreground label encoding in the video backbone model.

4.3. Qualitative Analysis

Fig. 7 shows the visual heat map of the cross attention in pattern VI, which exhibits the attention score of the intermediate layer in last triple-block. The red area in the heat map indicates a high attention degree and the query points are specified. The first row shows the full occlusion situation where the longer video sequences ($T > 2$) can effectively improve the robustness comparing to the tracking drift to the similar distractor object (person on white clothes) where limited frames are used ($T = 1$). The cross attention intensity shows that the model tends to aggregate the nearer temporal information than the farther one, validating that the over-long temporal extends do not provide more useful clues but redundancy. A more interesting fact lies in the second row: Two intermediate templates are copied from the same frame which still helps the model to successfully track the object, comparing to the baseline using the same intermediate template but only once. It is consistent with

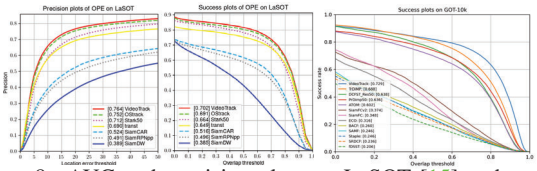


Figure 8. AUC and precision plots on LaSOT [15] and success plot on GOT-10k [24]. Better viewed with zooming in.

our observation that the appearance clue plays an more important role than motion clue in that matching with the same template double times enhances the discrimination ability.

4.4. Comparison to state-of-the-art

We compare our proposed VideoTrack with sota trackers on four popular VOT benchmarks: GOT-10k [24], LaSOT [15], TrackingNet [36] and UAV [35]. Please refer to the appendix for detailed description of benchmarks.

GOT-10k [24] : GOT-10k has the zero overlap of object classes between training and testing subset. We strictly follow the official GOT-10k protocol which forbids the external datasets for training. In Tab. 7 and Fig. 8, our approach obtains 72.9% AO score, significantly outperforming the sota pure-transformer trackers OStrack/SBT/Mixformer by 1.9%/2.5%/5.0%. VideoTrack also ranks the first in other two metrics: 81.9% in SR_{50} and 69.8% in SR_{75}.

LaSOT [15]: LaSOT is a large-scale long-term dataset, where temporal modelling is crucial. As shown in Tab. 8, VideoTrack achieves the top-rank AUC score (70.2%) and Precision score (76.4%), which surpasses the other three strong pair-wise Siamese trackers Ostrack/SBT/TransT for 1.1/3.5/5.3 points AUC score.

TrackingNet [36]: TrackingNet consists of 511 sequences for testing. Tab. 9 shows that, compared with sota models, VideoTrack ranks at the first in AUC score of 83.8% and normalized precision of 88.7%.

UAV123 [35] : UAV123 is a specific dataset for unmanned aerial vehicles, including 123 videos. In Tab. 10, the previous sota trackers such as OStrack [61], Mixformer [10], TransT [7], and Stark [58] are included, VideoTrack outperforms those methods by a considerable margin and achieves 69.7%/89.9% in AUC/Precision score.

5. Conclusion

In this work, we are the the first to utilize the video transformer backbone for VOT, which lifts the classic pair-wise Siamese matching to spatiotemporal domain. VideoTrack avoids labor-intensive temporal modelling modules and tedious online hyper-parameters, formulating a neat and conceptual simple framework to exploit temporal contexts. We conduct a systematic study on video transformer tracking, *e.g.* model architectures and temporal modelling methods. VideoTrack achieves promising results and may enlighten other template-matching tasks to choose video models.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 2, 3, 4

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 3, 4, 7

[4] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H S Torr. Fully-convolutional siamese networks for object tracking. In *ECCVW*, 2016. 1, 2, 8

[5] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 2, 8

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 3

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021. 3, 4, 8

[8] Siyuan Cheng, Bineng Zhong, Guorong Li, Xin Liu, Zhenjun Tang, Xianxian Li, and Jing Wang. Learning to filter: Siamese relation network for robust tracking. In *CVPR*, 2021. 2

[9] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021. 3

[10] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*, 2022. 1, 2, 3, 4, 8

[11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *CVPR*, 2017. 2, 8

[12] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019. 2, 8

[13] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020. 2, 8

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 3, 4, 6, 7

[15] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019. 6, 7, 8

[16] Xie Fei, Wang Chunyu, Wang Guangting, Cao Yue, Yang Wankou, and Zeng Wenjun. Correlation-aware deep tracking. In *CVPR*, 2022. 1, 2, 3, 4, 8

[17] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, 2021. 2, 8

[18] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019. 3

[19] Dongyan Guo, Yanyan Shao, Ying Cui, Zhenhua Wang, Liyan Zhang, and Chunhua Shen. Graph attention tracking. In *CVPR*, 2021. 8

[20] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*, 2020. 1, 2

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 6, 7

[22] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[23] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. In *ICVS*, 2008. 2

[24] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 43(5):1562–1577, 2019. 6, 7, 8

[25] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video instance segmentation using inter-frame communication transformers. *NIPS*, 2021. 3, 5

[26] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019. 1, 2, 8

[27] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 1, 2

[28] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022. 4

[29] Xin Li, Chao Ma, Baoyuan Wu, Zhenyu He, and Ming-Hsuan Yang. Target-aware deep tracking. In *CVPR*, 2019. 2

[30] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 6

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1, 3

[32] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022. 3, 5, 6, 7

[33] Alan Lukezic, Jiri Matas, and Matej Kristan. D3S-a discriminative single shot segmentation tracker. In *CVPR*, 2020. 8

[34] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *CVPR*, 2015. 2

[35] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016. 8

[36] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018. 6, 8

[37] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCV*, 2021. 3

[38] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *NIPS*, 2021. 6, 7

[39] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 3

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet Large scale visual recognition challenge. *IJCV*, 2015. 7

[41] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022. 6, 7

[42] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 7

[43] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 3

[44] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 3

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 3, 4

[46] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In *CVPR*, 2020. 2

[47] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, 2021. 1, 2, 8

[48] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *CVPR*, 2018. 2

[49] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 1, 3

[50] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 3

[51] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021. 3

[52] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 3

[53] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 1

[54] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *NIPS*, 2021. 4

[55] Fei Xie, Chunyu Wang, Guangting Wang, Yang Wankou, and Wenjun Zeng. Learning tracking representations via dual-branch fully transformer networks. In *ICCVW*, 2021. 1, 2, 8

[56] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 3

[57] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. SiamFC++: towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, 2020. 1, 2, 8

[58] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, 2021. 3, 4, 8

[59] Shusheng Yang, Xinggang Wang, Yu Li, Yuxin Fang, Jiemin Fang, Wenyu Liu, Xun Zhao, and Ying Shan. Temporally efficient vision transformer for video instance segmentation. In *CVPR*, 2022. 3, 5

[60] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *ECCV*, 2018. 1, 2

[61] Botao Ye, Hong Chang, Bingpeng Ma, and Shiguang Shan. Joint feature learning and relation modeling for tracking: A one-stream framework. *arXiv preprint arXiv:2203.11991*, 2022. 1, 2, 3, 8

[62] Bin Yu, Ming Tang, Linyu Zheng, Guibo Zhu, Jinqiao Wang, Hao Feng, Xuetao Feng, and Hanqing Lu. High-performance discriminative tracking with transformers. In *ICCV*, 2021. 2, 8

[63] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. Token shift transformer for video classification. In *ACMMM*, 2021. 3

[64] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, 2019. 1, 2

[65] Zhipeng Zhang, Yihao Liu, Xiao Wang, Bing Li, and Weiming Hu. Learn to match: Automatic matching network design for visual tracking. In *ICCV*, 2021. 8

[66] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 2019. 2

[67] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 8

[68] Linyu Zheng, Ming Tang, Yingying Chen, Jinqiao Wang, and Hanqing Lu. Learning feature embeddings for discriminant model based tracking. In *ECCV*, 2020. 2