Autoregressive Queries for Adaptive Tracking with Spatio-Temporal Transformers

Jinxia Xie[1,2,3], Bineng Zhong[1,2,3]*, Zhiyi Mo[3], Shengping Zhang[4], Liangtao Shi[1], Shuxiang Song[1], Rongrong Ji[5]
[1]Key Laboratory of Education Blockchain and Intelligent Technology Ministry of Education
[2]Guangxi Key Lab of Multi-Source Information Mining & Security
[1,2]Guangxi Normal University, Guilin 541004, China
[3]Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University
[4]School of Computer Science and Technology, Harbin Institute of Technology
[5]Media Analytics and Computing Lab, School of Informatics, Xiamen University
xie_jx@stu.gxnu.edu.cn, bnzhong@gxnu.edu.cn, zhiyim@gxuwz.edu.cn, s.zhang@hit.edu.cn
slt@stu.gxnu.edu.cn, songshuxiang@mailbox.gxnu.edu.cn, rrji@xmu.edu.cn

Abstract

The rich spatio-temporal information is crucial to capture the complicated target appearance variations in visual tracking. However, most top-performing tracking algorithms rely on many hand-crafted components for spatio-temporal information aggregation. Consequently, the spatio-temporal information is far away from being fully explored. To alleviate this issue, we propose an adaptive tracker with spatio-temporal transformers (named AQATrack), which adopts simple autoregressive queries to effectively learn spatio-temporal information without many hand-designed components. Firstly, we introduce a set of learnable and autoregressive queries to capture the instantaneous target appearance changes in a sliding window fashion. Then, we design a novel attention mechanism for the interaction of existing queries to generate a new query in current frame. Finally, based on the initial target template and learnt autoregressive queries, a spatio-temporal information fusion module (STM) is designed for spatio-temporal formation aggregation to locate a target object. Benefiting from the STM, we can effectively combine the static appearance and instantaneous changes to guide robust tracking. Extensive experiments show that our method significantly improves the tracker's performance on six popular tracking benchmarks: LaSOT, LaSOT$_{ext}$, TrackingNet, GOT-10k, TNL2K, and UAV123.Code and models will be https://github.com/orgs/GXNU-ZhongLab.

*Corresponding Author

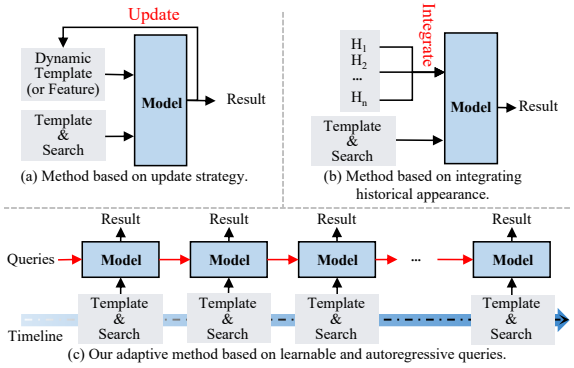


Figure 1. The comparison of three different tracking paradigms. (a) The trackers based on an updating strategy to update a dynamic template or feature. (b) The trackers based on integrating historical appearance. H represents historical appearance. (c) The proposed adaptive tracker with learnable and autoregressive queries.

1. Introduction

Visual object tracking(VOT) is a fundamental task in computer vision, which aims to estimate the position and shape of an arbitrary target in video sequences given its initial status. It has a wide range of applications in fields including robotic vision, video surveillance [10, 37], autonomous driving [14, 35], and various other domains. However, the tracking is often influenced by many factors, including camera movement, self-deformation, and external environment (occlusion or distractions from similar objects). And the target appearance always changing.

Due to the aforementioned challenges, mainstream tracking algorithms [7, 8, 18, 19, 45, 51] are difficult to effectively discriminate targets based on the static appear-

ance (initial template). Therefore few models [2, 11, 26, 38, 44, 48] explore spatio-temporal information to capture the appearance changes to improve their discriminative ability. These methods mine the spatio-temporal information in two main manners. The first one relies on an update strategy to update a new target appearance, as shown in Fig. 1(a). Some trackers [9, 11, 38] update a dynamic template to get a new target appearance. In addition, some trackers [2, 26] get the new target appearance by updating a state feature or motion feature. These methods based on an update strategy use a confidence score to update the template or feature in an interval. Even though these methods have achieved success, they require a manual design of update strategies and introduce hyperparameters (e.g., intervals, thresholds). The second approach to mining the spatio-temporal information is to integrate the historical target appearance, as illustrated in Fig. 1(b). These methods [17, 30, 46, 53] integrate historical appearance through some operation, including feature concatenation [46, 48], weighted sum [53], and memory networks [17]. Although these methods achieve a competitive performance, they require more computational resources and are more likely to lead to error accumulation.

To avoid the above issues, we propose an adaptive tracker (named AQATrack) with spatio-temporal transformers, which adopts simple autoregressive queries to effectively learn spatio-temporal information without cumbersome and customized components, as demonstrated in Fig. 1(c). Firstly, we use the HiViT [54] as the encoder, whose task is to learn outstanding spatial features of the target. Secondly, we design a decoder to mine and propagate spatio-temporal information across continuous frames. We introduce a set of learnable and autoregressive *target queries* to capture the instantaneous target appearance changes in a sliding window fashion. And *temporal attention* mechanism is used for the interaction of existing queries to generate a new query in the current frame. Finally, a spatio-temporal information fusion module (STM) is designed for spatio-temporal information aggregation to locate a target object without any hyperparameters. Benefiting from the STM, we can effectively combine the static and instantaneous target appearance changes to guide robust tracking. Detailed experiment shows that our method can effectively capture the target state changes and motion trends. Our main contributions are summarized as follows:

- To fully explore the spatio-temporal information, we propose an adaptive tracker to capture instantaneous appearance changes without any hand-designed components.
- In the proposed tracker, we introduce a set of learnable and autoregressive queries to capture the instantaneous target appearance changes in a sliding window fashion. A spatio-temporal information fusion module is designed to combine static appearance and instantaneous changes.
- Extensive experimental results demonstrate that our tracker achieves SOTA performance on six challenging benchmarks. In particular, AQATrack-256 and AQATrack-384 achieves 71.4% and 72.7% AUC score on long-term benchmarks LaSOT [15], respectively.

2. Related Work

Visual object tracking based on spatial features. Most trackers perform well by introducing a backbone with powerful spatial feature extraction capabilities from detection tasks or natural language processing (NLP) tasks. SiamFC [1] designed a siamese network framework using AlexNet [24] as the backbone network to extract features from template and search, which achieved good performance in speed and accuracy. Some trackers [8, 48] used ResNet [21] as a backbone and achieved excellent performance. In recent years, the transformer-based algorithms introduced to target recognition demonstrated astonishing global modeling capabilities. Therefore, visual object tracking also began to use transformers [40]. Initially, some trackers utilized an attention mechanism or transformer for feature extraction or fusion in tracking. Such as TransT [8] designed two modules based on the attention mechanism for feature interaction. STARK [48] uses the transformer structure as a fusion module. Later, due to the unstoppable charm of the transformer, some trackers [11, 26] took the transformer as a backbone. In addition, some researchers proposed full transformer-based trackers [11, 45, 51] which join feature extraction and fusion, greatly improving the performance.

Tracking combining spatio-temporal information. Spatio-temporal information is vital for the model to capture the target state changes and motion trends. Thus many mainstream studies [9, 11, 38, 46, 47] explored spatio-temporal information in visual object tracking. One usually used method is updating appearance representation. Many works [9, 11, 38] adopted a dynamic template to update the target appearance to capture changes, thus making the matching between template and search images more accurate. A few studies [2, 26] focused on learning a feature to describe the target's previous state or motion information. Another common method for exploring spatio-temporal information is integrating historical appearance. STMTrack [17] proposes a space-time memory network to make full use of historical information. UpdateNet [53] takes into account a set of historical appearances to estimate the optimal template for the next frame. However, most of these methods require the design with some artificial rules, and the spatio-temporal information is far from being fully explored. Recently, TCTrack[4]/TCTrack++[5] are proposed for temporal contexts in aerial tracking. They exploit spatio-temporal information on two levels: the extraction of features and the refinement of similarity maps. ARTrack[47] autoregressively predicts the current coordinates based on historical coordinates.

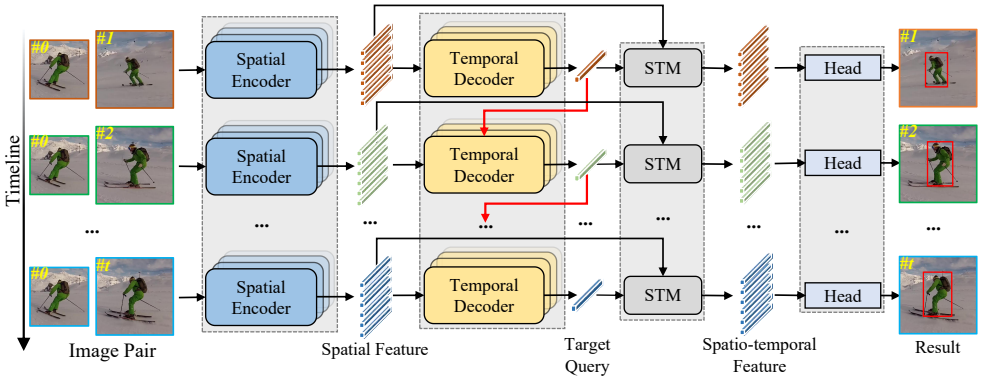


Figure 2. Overview of our framework. It mainly consists of four components, i.e., a spatial encoder for spatial features, a temporal decoder for learning an autoregressive target query that incorporates temporal information(with red arrows), a spatio-temporal feature fusion module(STM) designed for a spatio-temporal feature, and a prediction head.

The utilization of query. DETR [6] introduced the concept of *query* to detect different objects. Since then, many fields explored the usage of queries, i.e. video instance segmentation (VIS), multiple object tracking (MOTR), and Video object detection (VOD). Some algorithms [31, 43, 44, 52] use queries to help recognize the target, which collects information about the changing state of the object in the video clip. In the field of video segmentation, VisTR [43] adapts transformer [40] to VIS [50] and uses instance queries to obtain instance sequences from video clips. In the realm of multiple object tracking, MOTR [52] introduced *track query* to model the tracked instances in the entire video, transferring and updating it frame-by-frame for iterative predictions over time. Inspired by the usage of queries in so many excellent algorithms, we propose a decoder based on target queries to explore spatio-temporal information within video clips. To the best of our knowledge, we are the first to introduce queries in an autoregressive manner in single object tracking.

3. Method

In this section, we provide a detailed explanation of our proposed tracker. We start by giving a concise overview of our spatio-temporal tracking framework. Next, we delve into the specific components of our model, including the spatial encoder, temporal decoder with queries and temporal attention, as well as the spatio-temporal information fusion modules (STM). Finally, we introduce the head network and loss function used in the tracker.

3.1. Overview

As shown in Fig. 2, our proposed tracker primarily consists of a spatial encoder, a temporal decoder, and a spatio-temporal feature fusion operation module (STM). A pair of images are input into the encoder, including a template image $\mathbf{Z} \in \mathbb{R}^{H_z \times W_z \times 3}$ and a search region image $\mathbf{X} \in \mathbb{R}^{H_x \times W_x \times 3}$. The spatial encoder uses a hierarchical downsampling method for image processing and then learns outstanding representation features through attention mechanisms. The temporal decoder takes two inputs: the first one is a spatial feature from the spatial encoder, and the second one is some learnable and autoregressive queries. Its task is to learn the target appearance changes to better guide the expression of spatial features. In addition, we employed a spatio-temporal information fusion module (STM) to combine the static appearance and instantaneous appearance changes. At last, the output features of STM will be used for result prediction. We refer to the previous work and employ a center head network to predict the result.

3.2. Spatial Encoder

Many tracking algorithms [7, 20, 51] use a ViT [13] as the backbone, and its patch size for patch embedding is 16×16. However, we believe that conducting a large downsampling at once will weaken the correlation between different patches. So we use a gradual downsampling network as our spatial encoder, which is with a 4×4 patch size for patch embedding. A total of eight multi-layer perceptron (MLP) layers and two merging layers were used to achieve the goal of gradually downsampling. After the above operation, the obtained search tokens and template tokens are $f_z \in \mathbb{R}^{N_z \times D}$ and $f_x \in \mathbb{R}^{N_x \times D}$, respectively. Here, $N_z = H_z W_z / 16^2$, $N_x = H_x W_x / 16^2$, $D = 512$. Next, the template token and search token will be concatenated and fed into the N-layer encoder for spatial feature learning. The operation in our encoder can be described as the following equations:

$$\begin{aligned} f_{zx}^0 &= Concat(f_z, f_x), \\ f_{zx}^n &= Encoder(f_{zx}^{n-1}), n = 1...N, \\ f_{zx}^N &= LN(f_{zx}^N). \end{aligned} \tag{1}$$

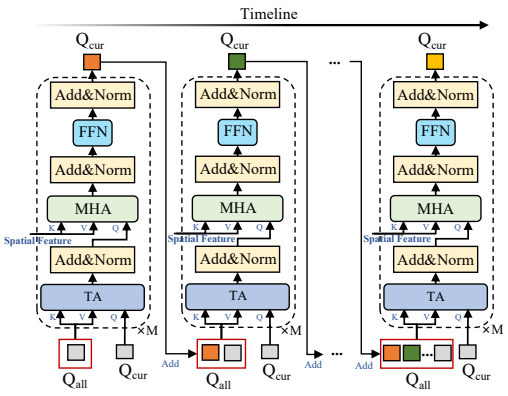


Figure 3. The structure of the temporal decoder is equipped with target query and temporal attention. Here FFN, MHA, and TA are feedforward neural networks, multi-head attention, and temporal attention, respectively. And Q represents the target query.

Refer to HiViT [54] for a more detailed design of our encoder.

3.3. Temporal Decoder

If only spatial features are used for tracking and prediction, it cannot effectively cope with challenges such as motion or interference from similar objects. Therefore, it is necessary to introduce temporal information to guide the expression of spatial features. The number of our temporal encoder layers is M, and each layer mainly includes the following parts: a temporal attention (TA) layer, a multi-head attention (MHA) layer, and a feedforward neural network (FFN) layer. There are two inputs to the temporal encoder, one is the spatial feature f_{zx}^{N} from the spatial encoder, and the other is temporal queries.

Here, let's review the multi-head attention mechanism, whose formula is as follows:

$$\begin{aligned} &\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\mathbf{H}_1,...,\mathbf{H}_{n_h})\mathbf{W}^O, \\ &\mathbf{H}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \\ &\text{Attention}(\mathbf{q},\mathbf{k},\mathbf{v}) = \text{Softmax}(\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{d_k}})\mathbf{v}. \end{aligned} \tag{2}$$

The following are learnable parameters: $\mathbf{W}_i^Q \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_m \times d_v}$, and $\mathbf{W}^O \in \mathbb{R}^{n_h \times d_m}$ are learnable parameter. In AQATrack, we employ multi-head attention with 8 heads, i.e., $n_h = 8$, $d_m = 512$, and $d_k = d_v = d_m/8 = 64$.

Target query. Inspired by works [31, 39, 52] in multiple object tracking (MOT) tasks, we introduced queries to capture the spatio-temporal information, called *target query*. To our knowledge, we are the first to introduce autoregressive queries for mining spatio-temporal information in the single object tracking (SOT) task. As shown in the Fig. 3, there are two types of query in our decoder: one is passed down from the previous frames called Q_{pre}, and the other is the query to be learned from the current frame called Q_{cur}. Q_{cur} can describe the state of the target in the current frame and integrate spatio-temporal information. The decoder input consists of Q_{all} and Q_{cur}, which can be describe as folows:

$$\begin{aligned} &Q_{all} = Concat(Q_{pre}, Q_{cur}), \\ &Q_{pre} = \begin{cases} Concat(Q_1,...,Q_{t-1}), & t < m \\ Concat(Q_{t-m+1},...,Q_{t-1}), & t \geq m \end{cases} \end{aligned} \tag{3}$$

where m is the length of spatio-temporal information. The Q_{cur} after the temporal decoder is propagated to the next frame of the video clip as one of the Q_{pre}. Thus, the temporal decoder learns spatio-temporal information in the form of sliding windows.

Temporal attention (TA). Self-attention gives equal attention to all queries, resulting in an inability to achieve pure temporal information from the Q_{pre}. In order to capture elaborate target state changes and motion trends, we design temporal attention for the interaction of existing queries to generate a new query in current frame. As shown in Fig. 3, Q is generated by Q_{cur}, K and V generated by Q_{all}. The calculation of temporal attention is the same as the Eq. (2).

3.4. Spatio-temporal Information Fusion Module

We used a simple and effective operation to fuse the spatial features and temporal information without introducing any parameters. Specifically, we first use a dot product to calculate the similarity $S \in \mathbb{R}^{N_x \times N_{temporal}}$ between search spatial features $f_x^N \in \mathbb{R}^{N_x \times D}$ and temporal information $f_{temporal} \in \mathbb{R}^{N_{temporal} \times D}$, where $D = 512$. The parameter-free operation can be described as the following equation:

$$S = f_x^N \odot f_{temporal}^\top, \tag{4}$$

where $\odot$ means *Dot product*. The similar scores S highlight the location where the target may be located. Then use it to enhance the expression of spatial feature f_x^N with *element-wise product*.

3.5. Head and Loss

We use a center-based head for predicting the centroid position and scale of the target. The outputs of the prediction head are the classification score map $P \in [0,1]^{\frac{H_x}{P} \times \frac{W_x}{P}}$, the size of the bounding box $B \in [0,1]^{2 \times \frac{H_x}{P} \times \frac{W_x}{P}}$, and the offset size $O \in [0,1)^{2 \times \frac{H_x}{P} \times \frac{W_x}{P}}$. The location with the highest classification score is taken as the target's location, and the final tracking result is calculated by combining the offset size and the bounding box size. We use focal [28] loss and GIoU [36] loss in the pre-

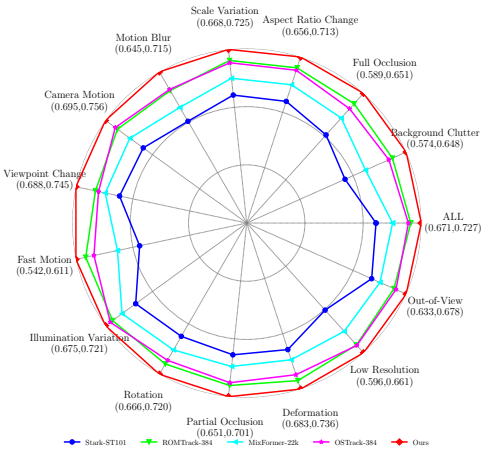


Figure 4. AUC scores of difference attributes on LaSOT[15]. Best viewed in color.

Table 1. Comparison of model parameters, FLOPs, and inference.

Model	Device	Speed(FPS)	MACs(G)	Params(M)	AUC(%)
SeqTrack-256[9]	2080Ti	40	66	89	69.9
STARK[48]	Tesla v100	31.7	18.5	43.4	67.1
AQATrack-256(ours)	Tesla v100	67.6	25.8	72	71.4
AQATrack-384(ours)	Tesla v100	44.2	58.3	72	72.7

diction head network, the total loss L calculation can be described as:

$$L = L_{cls} + \lambda_{iou} L_{iou} + \lambda_{L_1} L_1, \tag{5}$$

where $\lambda_{iou} = 2$ and $\lambda_{L1} = 5$ are the regularization parameters.

4. Experiments

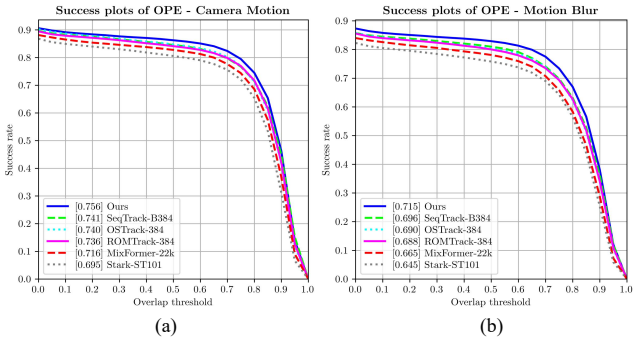
4.1. Implementation Details

Our algorithm uses Pytorch 1.9.0 in Python 3.8. Our tracker was trained and tested on 4 NVIDIA v100 GPUs.

Model variants. We present two variants of AQATrack with different configurations as follows:

- *AQATrack-256.* Template size:[128×128]; Search region size: [256×256];
- *AQATrack-384.* Template size:[192×192]; Search region size:[384×384].

AQATrack mainly includes three parts: spatial encoder, temporal decoder, and spatio-temporal information fusion module (STM). We use HiViT-Base [54] as the spatial encoder with a layer N of 20. In terms of the decoder, the number of layers M is 3, the hidden size is 256, the number of attention heads is 8, and the hidden size of the feed forward network (FFN) is 512. The number of temporal queries Q_{all} is four, of which three are previous temporal queries Q_{pre} and one is the current target query Q_{cur}.



(a) (b)

Figure 5. Success plots of one-pass evaluation (OPE) about camera motion and motion blur challenges on LaSOT[15]. Best viewed in color and zooming in.

Training strategy. Following traditional protocols, we trained our model on four datasets, namely: LaSOT [15], COCO [27], TrackingNet [33], and GOT-10k [23] (remove 1,000 videos as [48]). We test the GOT-10k with its training split to follow the protocol described in [23]. We use typical commonly used data augmentation methods, including horizontal flipping and brightness jittering. We train AQATrack with AdamW [29] optimizer, set the weight decay to 10^{-4}, the initial learning rate for the backbone to 4×10^{-5}, and other parameters to 4×10^{-4}. We trained the model for a total of 150 epochs with 60k image pairs, and the learning rate was decayed to 4×10^{-5} at the 120th epoch. For GOT-10k, we trained the model for a total of 100 epochs, and the learning rate decayed at the 80th epoch. To learn continuous spatio-temporal information, we used video-level sampling strategies in training. Specifically, we sample n video sequences each containing m template-search pairs (with the same template). So, the total batch size in each iteration is $n * m$. For the AQATrack-256, we set n and m to 4,8, respectively. For AQATrack-384, due to the limitations of GPU memory, n and m are 4 and 4 respectively.

Inference. We followed the common practice [8, 51, 55] and utilized the Hamming window to incorporate positional priors. We use a set of target queries provided by the previous frames in the inference process to mine instantaneous appearance changes. As demonstrated in the Tab. 1, we compared inference speed, MAC, and Params with state-of-the-art trackers. Our AQATrack can run in real-time at more than $65 fps$ which is more than twice as far as STARK [48].

4.2. Results and Comparisons

To demonstrate the effectiveness of our method, AQATrackis compared with the current SOTA trackers on six datasets, i.e., LaSOT [15], LaSOT$_{ext}$ [16], GOT-10K [23], TNL2K [42], UAV123 [32], TrackingNet [33].

LaSOT [15]. LaSOT consists of 280 videos for testing, serving as a challenging large-scale long-term tracking

Table 2. Performance comparisons with state-of-the-art trackers on the test set of LaSOT, $\text{LaSOT}_{\text{ext}}$, GOT-10K, TNL2K and UAV123. We add a symbol * over GOT-10k to indicate that the corresponding models are only trained with the GOT-10k training set. The top two results are highlighted with red and blue fonts, respectively.

Method	Source	LaSOT[15]			$\text{LaSOT}_{\text{ext}}$			GOT-10K*			TNL2K		UAV123
		AUC(%)	P_{norm}(%)	P(%)	AUC(%)	P_{norm}(%)	P(%)	AO(%)	$\text{SR}_{0.5}$(%)	$\text{SR}_{0.75}$(%)	AUC(%)	P(%)	AUC(%)
AQATrack-256	Ours	**71.4**	**81.9**	**78.6**	**51.2**	**62.2**	**58.9**	**73.8**	**83.2**	**72.1**	**57.8**	**59.4**	**70.7**
F-BDMTrack-256[49]	ICCV23	69.9	79.4	75.8	47.9	57.9	54.0	72.7	82.0	69.9	56.4	56.5	69.0
ROMTrack-256[3]	ICCV23	69.3	78.8	75.6	48.9	59.3	55.0	72.9	82.9	70.2	-	-	69.7
ARTrack-256[47]	CVPR23	**70.4**	79.5	**76.6**	46.4	56.5	52.3	73.5	82.2	70.9	**57.5**	-	67.7
SeqTrack-B256[9]	CVPR23	69.9	**79.7**	76.3	**49.5**	**60.8**	**56.3**	**74.7**	**84.7**	**71.8**	54.9	-	69.2
OSTrack-256[51]	ECCV22	69.1	78.7	75.2	47.4	57.3	53.3	71.0	80.4	68.2	54.3	-	68.3
VideoTrack[46]	CVPR23	70.2	-	76.4	-	-	-	72.9	81.9	69.8	-	-	69.7
SimTrack-B/16[7]	ECCV22	69.3	78.5	-	-	-	-	68.6	78.9	62.4	54.8	53.8	69.8
MixFormer-22k[11]	CVPR22	69.2	78.7	74.7	-	-	-	70.7	80.0	67.8	-	-	70.4
AiATrack-320[19]	ECCV22	69.0	79.4	73.8	-	-	-	69.6	80.0	63.2	-	-	**70.6**
STARK[48]	ICCV21	67.1	77.0	-	-	-	-	68.8	78.1	64.1	-	-	-
AutoMatch[56]	ICCV21	58.3	-	59.9	-	-	-	65.2	76.6	54.3	47.2	43.5	-
TransT [8]	CVPR21	64.9	73.8	69.0	-	-	-	67.1	76.8	60.9	50.7	51.7	69.1
TrDiMP[41]	CVPR21	63.9	-	61.4	-	-	-	67.1	77.7	58.3	-	-	67.5
Ocean [55]	ECCV 20	56.0	65.1	56.6	-	-	-	61.1	72.1	47.3	38.4	37.7	-
SiamPRN++[25]	CVPR19	49.6	56.9	49.1	34.0	41.6	39.6	51.7	61.6	32.5	41.3	41.2	61.0
ECO [12]	ICCV 17	32.4	33.8	30.1	22.0	25.2	24.0	31.6	30.9	11.1	32.6	31.7	53.5
MDNet [34]	CVPR16	39.7	46.0	37.3	27.9	34.9	31.8	29.9	30.3	9.9	38.0	37.1	52.8
SiamFC [1]	ECCVW16	33.6	42.0	33.9	23.0	31.1	26.9	34.8	35.3	9.8	29.5	28.6	46.8
Some Trackers with Higher Resolution													
OSTrack-384[51]	ECCV22	71.1	81.1	77.6	50.5	61.3	57.6	73.7	83.2	70.8	55.9	-	70.7
SeqTrack-B384[9]	CVPR23	71.5	81.1	77.8	50.5	61.6	57.5	74.5	**84.3**	71.4	56.4	-	68.6
ARTrack-384[47]	CVPR23	**72.6**	**81.7**	**79.1**	**51.9**	62.0	58.5	**75.5**	**84.3**	**74.3**	**59.8**	-	70.5
ROMTrack-384[3]	ICCV23	71.4	81.4	78.2	51.3	**62.4**	**58.6**	74.2	**84.3**	72.4	-	-	70.5
F-BDMTrack-384[49]	ICCV23	72.0	81.5	77.7	50.8	61.3	57.8	75.4	**84.3**	72.9	57.8	**59.4**	**70.9**
AQATrack-384	Ours	**72.7**	**82.9**	**80.2**	**52.7**	**64.2**	**60.8**	**76.0**	**85.2**	**74.9**	**59.3**	**62.3**	**71.2**

Table 3. Performance comparisons with state-of-the-art trackers on the test set of TrackingNet. The top two results are highlighted with red and blue fonts respectively.

	SiamFC [1]	ECO [12]	SiamRPN++ [25]	TransT [8]	STARK [48]	MixFormer-22k [11]	AiATrack [19]	OSTrack [51]	ARTrack [47]	SeqTrack [9]	F-BDMTrack [49]	AQATrack-256 ours	AQATrack-384 ours
AUC(%)	57.1	55.4	73.3	81.4	82.0	83.1	82.7	83.9	**85.1**	83.9	83.7	83.8	**84.8**
P_{Norm}(%)	66.3	61.8	80.0	86.7	86.9	88.1	87.8	88.5	**89.1**	88.8	88.3	88.6	**89.3**
P(%)	53.3	49.2	69.4	80.3	-	81.6	80.4	83.2	**84.8**	83.6	82.6	83.1	**84.3**

benchmark. As illustrated in Tab. 2, we compare the results of the AQATrack with the previous SOTA trackers. The results show that AQATrack-256 achieved 71.4% AUC on LaSOT [15], significantly outperforming other trackers at the same resolution. And AQATrack-384 with an AUC score of 72.7% surpassing all other trackers without bells and whistles. As shown at Fig. 4, AQATrack demonstrate competitive performance in all challenges on LaSOT. Particularly, Fig. 5 demonstrates AQATrack outperforms previous SOTA trackers in camera motion and motion blur challenges. The outstanding performance on the LaSOT benchmark mentioned above proves the effectiveness of our approach in mining spatio-temporal information.

LaSOT$_{ext}$ [16]. LaSOT_{ext} expands upon the LaSOT [15] dataset by adding 150 videos. These introduced sequences pose significant challenges due to the presence of numerous interference of similar objects in the videos. The results presented in Tab. 2 demonstrate that our AQATrack-256 outperforms all other trackers by a substantial margin, achieving the highest P_{norm} score of 62.2% and surpassing ARTrack [47] by 1.4%. Additionally, our higher resolution model AQATrack-384 also significantly outperforms previous trackers in all three metrics. This establishes a new state-of-the-art on LaSOT_{ext}, indicating that our tracker has a robust discrimination ability against similar distractors.

GOT-10k [23]. In the GOT-10k dataset's test set, a one-shot tracking rule is applied. This rule mandates that trackers are solely trained on the GOT-10k training split, and there is no overlap in object classes between the train and test splits. We adhere to this protocol for training our model and evaluate the results by submitting them to the official evaluation server. As indicated in Tab. 2, AQATrack-384 and AQATrack-256 outperform most of the previous trackers. This highlights our trackers' ability to perceive the target's changing state. The exceptional performance on this one-shot benchmark underscores the effectiveness of AQATrack in extracting spatio-temporal information for unseen classes in an adaptive and autoregression manner.

Table 4. Ablation study for important components. Blank denotes the component is used by default, while ✗ represents the component is removed. Performance is evaluated on LaSOT[15].

#	Decoder	STM	TA	Q_{all}	Autoregression	AUC(%)	P_{norm}(%)	P(%)
1	✗	✗				70.5	80.7	77.5
2			✗	✗		70.8	80.9	77.6
3				✗	✗	70.5	80.6	77.4
4						**71.4**	**81.9**	**78.6**

TNL2K [42]. TNL2K is a tracking dataset consisting of 700 videos available for testing. AQATrack-256 achieved a new state-of-the-art score of 57.8% in AUC at the same resolution, as shown in the Tab. 2. AQATrack-384 also demonstrated outstanding performance, with AUC score of 59.3%.

UAV123 [32]. UAV123 is a dataset for unmanned aerial vehicles, comprising 123 videos. In the Tab. 2, we present the results, including our model and previous state-of-the-art trackers such as OSTrack, MixFormer, TransT, and STARK, among others. Our model significantly outperforms these methods, achieving two AUC scores of 70.7% and 71.2% with a considerable margin.

TrackingNet [33]. The TrackingNet benchmark comprises 511 testing sequences. As shown in Tab. 3, AQATrack-384 demonstrated competitive performance compared to previous state-of-the-art trackers. AQATrack gets the best P_{norm}, surpassing the previous best-performing tracker ARtrack [47].

4.3. Ablation Study and Analysis

To demonstrate the effectiveness of our proposed spatio-temporal information learning method, we conducted some ablation study using AQATrack-256 as the baseline on LaSOT [15].

Spatial encoder. Recently, the vanilla ViT-Base [13] model pretrained with MAE [22], has been frequently used as a backbone for feature extraction and fusion. For a fair comparison, we conducted a study using ViT as the spatial encoder in AQATrack-256, and the results as shown in Tab. 5. The ViT-based AQATrack improved by 0.5% in terms of success compared to only ViT, which indicates the effectiveness of the proposed temporal decoder with learnable and autoregressive queries. To avoid damaging the integrity of spatial information, we use a hierarchical transformer (HiViT [54]) as the encoder. The HiViT-based AQATrack improved by 0.9% compared to only HiViT. In summary, the HiViT-based AQATrack-256 outperforms the ViT-based AQATrack-256 by 0.8% in AUC. It also shows that our spatial encoder can better capture the spatial features of targets. This is attributed to the fact that, unlike ViT where images are downsampled 16 times simultaneously, HiViT progressively downsamples images before entering the attention block. So it facilitates the effective preservation of spatial features.

Table 5. The effectiveness of our method using other encoders on LaSOT[15].

Encoder	Temporal Decoder	AUC(%)	P_{norm}(%)	P(%)
ViT[13]	-	69.1	79.1	75.1
	✓	69.6	79.5	75.6
HiViT[54]	-	70.5	80.7	77.5
	✓	**71.4**	**81.9**	**78.6**

Table 6. Experimental studies on the number of decoder layers.

Number of Decoder Layers(M)	AUC(%)	P_{norm}(%)	P(%)
1	70.5	80.6	77.2
3	**71.4**	**81.9**	**78.6**
6	71.1	81.2	77.9

Temporal decoder. The temporal decoder is a key module for instantaneous spatio-temporal information learning. To demonstrate the effectiveness of this module, we tested the impact of different numbers of temporal decoder layers (M) on our method. When M is 0, it is equivalent to removing the temporal decoder and STM from our network, leaving a spatial encoder, degenerated into a structure similar to OSTrack [51]. As demonstrated in the Tab. 4 (#1), the tracker only has a spatial encoder and achieved an AUC score of 70.5% on the LaSOT benchmark, which is 0.9% lower than the AQATrack that uses both spatio-temporal information (as shown in Tab. 4 (#4)). This proves that learning solely based on spatial features cannot accurately track the target. As illustrated in Tab. 6, when the number of decoder layers increases from 1 to 3, the decoder learns richer temporal information, resulting in better performance. The three variants in Tab. 6 are trained using the same configuration, except for the number of layers. When the number of layers M is 6, AQATrack's performance slightly decreases.

Temporal queries and length of spatio-temporal information. We introduce a target query to describe the state of the target. The input of the decoder is the target query of the previous m-1 frames and the current frame, which means that the current frame's state references the previous frames. So the length of spatio-temporal information obtained by the model is m. We investigate the impact of different spatio-temporal information lengths on AQATrack performance. As shown in Tab. 7 (#1, #2, #3), the performance of the model increases as the spatio-temporal range(m) increases from 1 to 4. This proves that when there are more frames for model reference, more target state changes can be obtained, thereby achieving better performance. However, when m is 8 and 16, the performance of the model slightly decreases. This is because the number of video sequences(n) used in each iteration is too small, resulting in a lack of generalization ability.

Table 7. Influence of the length of spatio-temporal information.

Batchsize	m	n	AUC(%)	P_{norm}(%)	P(%)
32	1	32	70.5	80.6	77.4
32	2	16	70.8	80.8	77.5
32	4	8	**71.4**	**81.9**	**78.6**
32	8	4	70.6	80.7	77.7
32	16	2	70.2	80.5	77.3

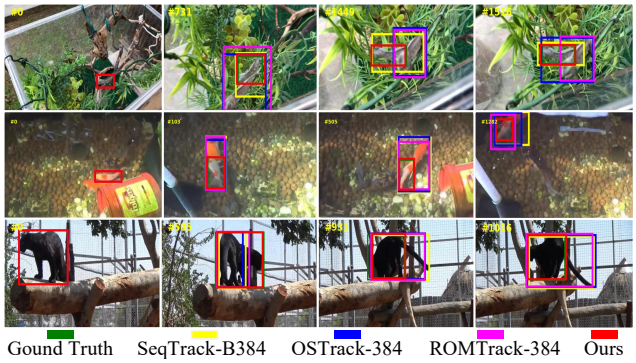


Figure 6. Comparison tracking result with other three SOTA trackers on LaSOT[15] benchmark.

Temporal attention (TA). To demonstrate temporal attention's key role in collecting target state change information across frames, we conducted experiments using self-attention as a substitute. As a result, the AUC, P_{norm}, and P obtained by the self-attention-based tracker on LaSOT [15] were 70.7%, 80.9%, and 77.8%, respectively, which are lower than tracker 0.9% on AUC. This experiment proves that our designed temporal attention is more effective in capturing the target motion trend across frames.

Spatial and temporal fusion operation module(STM). In AQATrack, STM is a simple and effective module that does not require learning parameter weights. It can enhance important regions using simple arithmetic operations. We attempted to use a self-attention layer (i.e., one block of ViT [13]) to fuse temporal information and spatial features, and the result was a 70.5% (-0.9% compared to STM) AUC score on the LaSOT benchmark.

Visualization and qualitative comparison. To intuitively demonstrate the effectiveness of our proposed continuous spatio-temporal modeling approach in occluded scenes, we visualized the tracking results of AQATrack and previous three SOTA models (i.e., SeqTrack [9], OSTrack [51], ROMTrack [3]). As shown in the Fig. 6, due to our tracker's excellent continuous spatio-temporal modeling ability to capture more delicate and subtle changes in the target state, our tracker achieved the most accurate tracking in these occlusion sequences. In addition, to demonstrate the effectiveness of temporal information, we also visualized the attention maps of the spatial encoder and temporal decoder, as demonstrated in Fig. 7. It can be seen that under the guidance of temporal information, the model pays more attention to the location of the target.

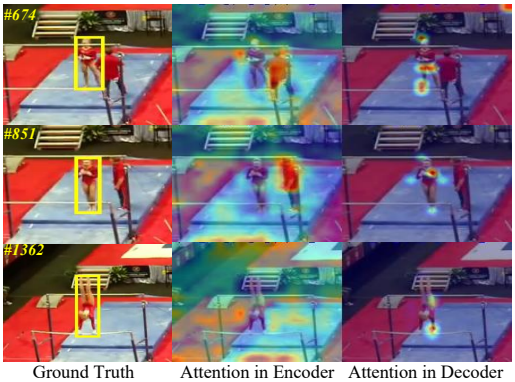


Figure 7. The comparison attention map in the spatial encoder and the temporal decoder on LaSOT[15]. The first column is the ground truth, the second column is the self-attention in the encoder, and the third column is the multi-head attention(MHA) in the decoder.

5. Conclusion

We propose AQATrack for continuous spatio-tmeporal information modeling from a novel perspective. We design a temporal decoder with temporal attention and temporal queries, which captures the motion trend to discriminate the target from similar objects. Empirical evaluation shows that our method is effective and achieves competitive performance compared to previous SOTA trackers.

Limitation. Our proposed method uses learnable and autoregressive target queries to capture the concentrated spatio-temporal information, reducing the unnecessary background interference introduced by some methods such as dynamic templates. So we believe that AQATrack with longer spatio-temporal information can exhibit stronger performance. However, due to the limitations of GPU memory, the modeling of more long-term spatio-temporal information in AQATrack has not been thoroughly explored.

Acknowledgement. This work is supported by the Project of Guangxi Science and Technology (No.2022GXNSFDA035079), the National Natural Science Foundation of China (No.U23A20383, U21A20474, and 62272134), the Guangxi "Young Bagui Scholar" Teams for Innovation and Research Project, the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, and the Guangxi Talent Highland Project of Big Data Intelligence and Application.

References

[1] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, pages 850–865, 2016. 2, 6

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, pages 205–221, 2020. 2

[3] Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. *CoRR*, abs/2308.05140, 2023. 6, 8

[4] Ziang Cao, Ziyuan Huang, Liang Pan, Shiwei Zhang, Ziwei Liu, and Changhong Fu. Tctrack: Temporal contexts for aerial tracking. In *CVPR*, pages 14778–14788, 2022. 2

[5] Ziang Cao, Ziyuan Huang, Liang Pan, Shiwei Zhang, Ziwei Liu, and Changhong Fu. Towards real-world visual tracking with temporal contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 3

[7] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In *ECCV (22)*, pages 375–392, 2022. 1, 3, 6

[8] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, pages 8126–8135, 2021. 1, 2, 5, 6

[9] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *CVPR*, 2023. 2, 5, 6, 8

[10] Linsong Cheng, Jiliang Wang, and Yinghui Li. Vitrack: Efficient tracking on the edge for commodity video surveillance systems. *IEEE Transactions on Parallel and Distributed Systems*, 33(3):723–735, 2022. 1

[11] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*, pages 13598–13608, 2022. 2, 6

[12] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, pages 6931–6939, 2017. 6

[13] 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. 3, 7, 8

[14] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9710–9719, 2021. 1

[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*, pages 5374–5383, 2019. 2, 5, 6, 7, 8

[16] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. Lasot: A high-quality large-scale single object tracking benchmark. *Int. J. Comput. Vis.*, pages 439–461, 2021. 5, 6

[17] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, pages 13774–13783, 2021. 2

[18] Zhihong Fu, Zehua Fu, Qingjie Liu, Wenrui Cai, and Yunhong Wang. Sparsett: Visual tracking with sparse transformers. *arXiv preprint arXiv:2205.03776*, 2022. 1

[19] Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *ECCV (22)*, pages 146–164, 2022. 1, 6

[20] Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18686–18695, 2023. 3

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 15979–15988, 2022. 7

[23] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5): 1562–1577, 2021. 5, 6

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 2

[25] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4282–4291, 2019. 6

[26] Liting Lin, Heng Fan, Zhipeng Zhang, Yong Xu, and Haibin Ling. Swintrack: A simple and strong baseline for transformer tracking. *Advances in Neural Information Processing Systems*, 35:16743–16754, 2022. 2

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 5

[28] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017. 4

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[30] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8731–8740, 2022. 2

[31] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 4

[32] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *ECCV*, pages 445–461, 2016. 5, 7

[33] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, pages 310–327, 2018. 5, 7

[34] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016. 6

[35] Chinthaka Premachandra, Shohei Ueda, and Yuya Suzuki. Detection and tracking of moving objects at road intersections using a 360-degree camera for driver assistance and automated driving. *IEEE Access*, 8:135652–135660, 2020. 1

[36] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 658–666. Computer Vision Foundation / IEEE, 2019. 4

[37] Ahsan Shehzed, Ahmad Jalal, and Kibum Kim. Multi-person tracking in smart surveillance system for crowd counting and normal/abnormal events detection. In *2019 International Conference on Applied and Engineering Mathematics (ICAEM)*, pages 163–168, 2019. 1

[38] Zikai Song, Run Luo, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Compact transformer tracker with correlative masked modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 2

[39] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv: 2012.15460*, 2020. 4

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2, 3

[41] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, pages 1571–1580, 2021. 6

[42] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *CVPR*, pages 13763–13773, 2021. 5, 7

[43] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8741–8750, 2021. 3

[44] Junfeng Wu, Yi Jiang, Wenqing Zhang, Xiang Bai, and Song Bai. Seqformer: a frustratingly simple model for video instance segmentation. *arXiv preprint arXiv:2112.08275*, 2021. 2, 3

[45] Fei Xie, Chunyu Wang, Guangting Wang, Yue Cao, Wankou Yang, and Wenjun Zeng. Correlation-aware deep tracking. In *CVPR*, pages 8741–8750, 2022. 1, 2

[46] Fei Xie, Lei Chu, Jiahao Li, Yan Lu, and Chao Ma. Videotrack: Learning to track objects via video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22826–22835, 2023. 2, 6

[47] Wei Xing, Bai Yifan, Zheng Yongchao, Shi Dahu, and Gong Yihong. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9697–9706, 2023. 2, 6, 7

[48] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, pages 10428–10437, 2021. 2, 5, 6

[49] Dawei Yang, Jianfeng He, Yinchao Ma, Qianjin Yu, and Tianzhu Zhang. Foreground-background distribution modeling transformer for visual object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10117–10127, 2023. 6

[50] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3

[51] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *ECCV (22)*, pages 341–357, 2022. 1, 2, 3, 5, 6, 7, 8

[52] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: end-to-end multiple-object tracking with transformer. In *ECCV (27)*, pages 659–675, 2022. 3, 4

[53] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, pages 4009–4018, 2019. 2

[54] Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *International Conference on Learning Representations*, 2023. 2, 4, 5, 7

[55] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, pages 771–787, 2020. 5, 6

[56] Zhipeng Zhang, Yihao Liu, Xiao Wang, Bing Li, and Weiming Hu. Learn to match: Automatic matching network design for visual tracking. In *ICCV*, pages 13319–13328, 2021. 6