Robust Tracking via Mamba-based Context-aware Token Learning

Jinxia Xie[1,2], Bineng Zhong[1,2*], Qihua Liang[1,2], Ning Li[1,2], Zhiyi Mo[3], Shuxiang Song[1,2]

[1]Key Laboratory of Education Blockchain and Intelligent Technology Ministry of Education, Guangxi Normal University, Guilin 541004, China

[2]Guangxi Key Lab of Multi-Source Information Mining & Security, Guangxi Normal University, Guilin 541004, China

[3]Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University, Wuzhou 543002, China

xie_jx@stu.gxnu.edu.cn, bnzhong@gxnu.edu.cn, qhliang@gxnu.edu.cn, ningli65536@mailbox.gxnu.edu.cn, zhiyim@gxuwz.edu.cn, songshuxiang@mailbox.gxnu.edu.cn

Abstract

How to make a good trade-off between performance and computational cost is crucial for a tracker. However, current famous methods typically focus on complicated and time-consuming learning that combining temporal and appearance information by input more and more images (or features). Consequently, these methods not only increase the model's computational source and learning burden but also introduce much useless and potentially interfering information. To alleviate the above issues, we propose a simple yet robust tracker that separates temporal information learning from appearance modeling and extracts temporal relations from a set of representative tokens rather than several images (or features). Specifically, we introduce one track token for each frame to collect the target's appearance information in the backbone. Then, we design a mamba-based Temporal Module for track tokens to be aware of context by interacting with other track tokens within a sliding window. This module consists of a mamba layer with autoregressive characteristic and a cross-attention layer with strong global perception ability, ensuring sufficient interaction for track tokens to perceive the appearance changes and movement trends of the target. Finally, track tokens serve as a guidance to adjust the appearance feature for the final prediction in the head. Experiments show our method is effective and achieves competitive performance on multiple benchmarks at a real-time speed.

Code — https://github.com/GXNU-ZhongLab/TemTrack

Introduction

Visual tracking is one of the fundamental tasks in computer vision, widely used in many fields, such as mobile robotics(Pereira et al. 2022), video surveillance(Cheng, Wang, and Li 2022; Shehzed, Jalal, and Kim 2019), and autonomous driving(Premachandra, Ueda, and Suzuki 2020). However, there are many challenges during the tracking process that affect the robustness of the trackers, such as occlusion, drastic appearance changes, and deformation.

Therefore, many methods(Li et al. 2019; Xu et al. 2020; Chen et al. 2020; Fu et al. 2022; Song et al. 2023; Xie et al. 2024; Hu et al. 2024a) are proposed and attempt to overcome the above challenges. These methods can be roughly divided into two types: trackers focused more on appearance and trackers combined appearance with temporal information. For the first type of trackers(Ye et al. 2022; Chen et al. 2021), they focus on building a more robust appearance model via a stronger backbone or a more efficient feature fusion method for template and search image. However, it's difficult for this type of trackers(Bertinetto et al. 2016; Ye et al. 2022; Chen et al. 2022; Gao, Zhou, and Zhang 2023) to recognize the correct target when facing severe appearance changes or interference from similar objects. Recently, the visual tracking community pay more attention to extract temporal context to mitigate the above difficulty. Many second type of trackers(Zheng et al. 2024; Bai et al. 2024; Xie et al. 2023; Cai, Liu, and Wang 2024; Cui et al. 2022, 2024) arise, combining appearance and temporal information. Thanks to introducing temporal information, these trackers perceive the appearance changes and motion trends of the target, and achieve competitive performance. However, they usually focus on complicated and time-consuming learning, inputting more images (or features), and leading the model more cumbersome and clumsy. Specifically, they(Yan et al. 2021; Chen et al. 2023; Lin et al. 2022) need to select additional im-

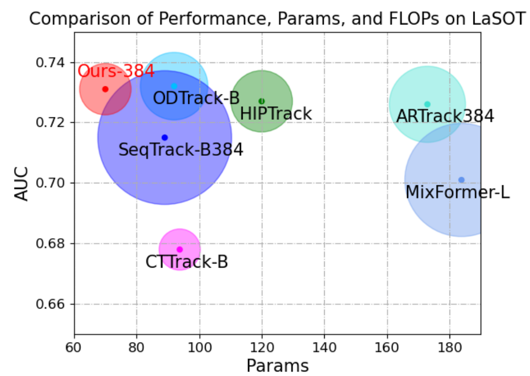


Figure 1: Comparison of AUC, Params, and FlOPs of recent SOTA trackers. The trackers that lower Params and higher AUC is closer to top-left corner. The size of a circles represents the tracker's FLOPs.

[*]Corresponding Author

ages besides one template and one search image, which requires controlling thresholds or manually crafting components for the selection strategy. These processes are tedious and not flexible. Furthermore, even when simple methods are employed for selecting images (or features), the large input size can significantly increase the computational resource and learning burden, and lead to heavy training costs. For instance, SeqTrack(Chen et al. 2023) inputs two templates with the same size as the search image, and its number of floating point operations (FLOPs) is 148G, which is nearly three times our tracker (55.7G), as shown in table 1. And ODTrack(Zheng et al. 2024) input three templates and one search image, which is time-consuming for model learning. Finally, increasing the number of images may introduce more useless or potentially interfering information leading to a suboptimal tracking result. The comparison with recent context-aware trackers of params, FLOPs, and performance on LaSOT(Fan et al. 2019) is shown in fig. 1.

To make a good trade-off between performance and computational cost, we propose a simple and efficient context-aware tracker, named TemTrack, which separates temporal information learning from appearance modeling and learns contextual information from a set of track tokens instead of images. In this way, it can alleviate the computational source and learning burden caused by inputting too many images, and the backbone network can focus more on learning the target appearance and modeling the relationship between templates and search images. Specifically, we introduce a track token for each frame and feed it into the backbone alongside template and search tokens. Each track token is responsible for collecting the appearance information of the target. After the backbone, each token contains the appearance information of the target in that frame. Then, we set a sliding window with a size of m. The track tokens in the sliding window are fed into a mamba-based Temporal Module for temporal context learning. This module consists of a mamba layer with a autoregressive characteristic and a cross-attention layer with strong global perception, which ensures sufficient interaction for track tokens to perceive the appearance changes and movement trends of the target. After interaction with the other tokens, the track token contains temporal information. Finally, we use the track token to adjust search features through simple operations, and then search features are fed into the head to predict the target's position and size. To summarize, the main contributions of this work are as follows:

- To make a good trade-off between performance and computational cost, we propose a simple but robust tracker, which separates temporal information learning from appearance modeling, extracting temporal relations from a set of representative tokens in a sliding window fashion.
- We develop an efficient mamba-based module for modeling contextual information, named Temporal Module. This module consists of mamba and attention mechanism, combining long sequence modeling and global perception capabilities.
- We conduct detailed experiments to verify the effectiveness of our temporal context information modeling method. The results demonstrate our method achieve a new state-of-art on multiple benchmarks.

Related Work

Trackers Focusing on Appearance Modeling. With the development of deep learning and the introduction of attention mechanisms, significant progress has been made in visual object tracking. Many trackers(Ye et al. 2022; Song et al. 2022; Hu et al. 2024b) focus more on appearance modeling, use a powerful backbone, and design a more effective module for feature confusion. SiamFC(Bertinetto et al. 2016) based AlexNet(Krizhevsky, Sutskever, and Hinton 2012) design a Siamese network to extract features and use fully-convolutional deep networks to fuse the feature. TransT(Chen et al. 2021) uses ResNet(He et al. 2016) as the backbone and introduces the attention to design a correlation module. Thanks to Swin Transformer(Liu et al. 2021) as the backbone, SwinTrack(Lin et al. 2022) achieves outstanding performance. One of the most successful trackers is OSTrack(Ye et al. 2022), which uses ViT(Dosovitskiy et al. 2021) as the backbone and proposes a simple yet effective one-stream tracking paradigm. Thus some trackers(Chen et al. 2022; Shi et al. 2024) adopt the one-stream paradigm and introduce other strong transformer variants as backbone, making significant progress. Our tracker adopts the great one-stream paradigm to model the appearance.

Trackers Combining Appearance and Temporal Context. Temporal information captures the appearance changes and motion patterns of targets, playing a crucial role in enhancing robustness against drastic appearance changes and interference from similar objects. So many trackers(Yan et al. 2021; Gao et al. 2022; Xue et al. 2024; Bai et al. 2024) combine appearance and temporal information to help trackers achieve more accurate tracking. Most trackers introduce the temporal information by updating a dynamic template image, which requires controlling thresholds or manually crafting components, such as MixFormer(Cui et al. 2022), CTTrack(Song et al. 2023), and SeqTrack(Chen et al. 2023). In addition, UpdateNet(Zhang et al. 2019) estimates an optimal template from several images for the next frame. STMTrack(Fu et al. 2021) uses a memory network to integrate historical features. VideoTrack(Xie et al. 2023) mines temporal information from video clips. Some trackers(Cao et al. 2023; Shi et al. 2024; Xie et al. 2024; Zheng et al. 2024) transmit temporal context to enhance the tracker's ability to distinguish targets. Although the above trackers achieve good performance, these models are usually more complex due to the need to design strategies to select images and withstand more learning and computational burden from inputting more images (or features). So we design a simple yet robust tracker with less computational cost, without updating strategy or inputting more images.

Mamba in Visual Task. Recently, the mamba with autoregressive characteristic become famous for its linear complexity and is introduced into many visual tasks. In upstream tasks, Vmamba(Liu et al. 2024) constructs a hierarchical vision model based on mamba with a four-direction scanning strategy. Vision Mamba(Zhu et al. 2024) proposes a bidirec-

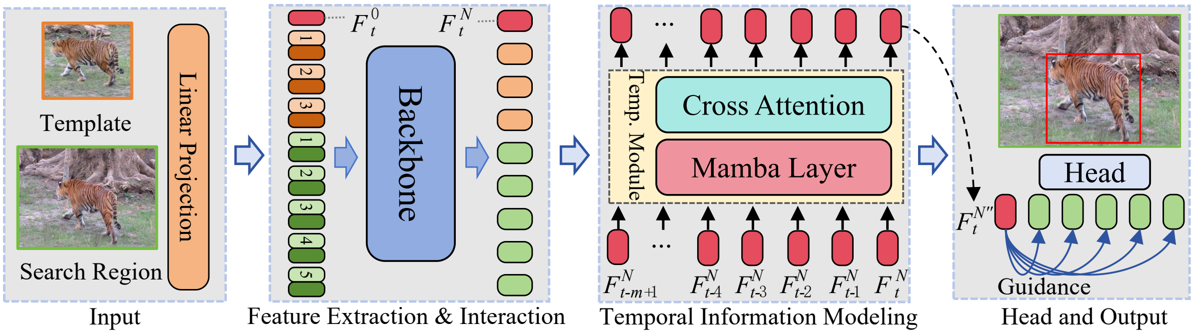


Figure 2: Overview of the proposed tracker TemTrack. The tracker's workflow is depicted from left to right, including feature extraction & interaction, temporal information modeling, and the final head stage. First, we add a track token $\boldsymbol{F}_t$ concatenating with template and search tokens to gather the target's appearance in the backbone. Furthermore, we develop a Temporal Module to associate track tokens to dig temporal information. Finally, the track tokens guide the adjustment of the search features to achieve more accurate predictions in the head network.

tional state space model referred to ViT's(Dosovitskiy et al. 2021) pipeline. LocalMamba(Huang et al. 2024) incorporates local inductive biases to enhance visual mamba models. In medical object segmentation, numerous studies adopt mamba-based models, such as U-Mamba(Ma, Li, and Wang 2024) and SegMamba(Xing et al. 2024). So many success models demonstrate mamba's outstanding long-sequence processing capabilities. In this work, we integrate mamba into Temporal Module to ensure sufficient interaction between track tokens.

Our Method

This section offers a concise and lucid description of the proposed robust temporal tracker, called TemTrack. First, we describe the tracking framework of TemTrack. Then, we introduce the main components, including a backbone and the Temporal Module. Finally, we briefly describe the guidance from track token to appearance, head, and loss function.

Overview

The framework of the TemTrack is demonstrated in fig. 2, whose main components are a strong backbone, a mamba-based Temporal Module, and a head. The input for the tracker is a pair of images, namely one template image $\mathbf{Z} \in \mathbb{R}^{h_z \times W_z \times 3}$ and one search image $\mathbf{X} \in \mathbb{R}^{h_x \times W_x \times 3}$. These two images are embedded and then concatenate with a track token to be fed into the backbone. The track token is one of the key components of TemTrack, whose responsibility is to gather the target's appearance from the image tokens in the backbone and learn the temporal context in the Temporal Module. Before the head, the appearance (search features) is adjusted by track token with temporal information, and fed into the head for the final prediction.

Feature Extraction and Relation Modeling

OStrack(Ye et al. 2022) proves that joint feature extraction and relationship modeling can enable sufficient interaction between templates and search features. Trackers(Gao, Zhou, and Zhang 2023) modeled using this approach can greatly improve their ability to discriminate targets. They usually use Vanilla ViT(Dosovitskiy et al. 2021) as a backbone to complete the above goals. ViT embeds the images to patches with size 16 × 16 at once, which loses a lot of information about adjacent patches(Xie et al. 2024). To avoid this issue, we choose Fast-iTPN(Tian et al. 2024) as the backbone, which performs downsampling twice via two merge layers before global attention. After downsampling, the features shape of the template and search are $\boldsymbol{F}_z^0 \in \mathbb{R}^{N_z \times D}$ and $\boldsymbol{F}_x^0 \in \mathbb{R}^{N_x \times D}$, respectively. Here, $N_z = h_z W_z/16^2$, $N_x = h_x W_x/16^2$, $D = 512$. So the patch size after downsampling is the same as other trackers, both are 16 × 16. To learn temporal information with a small cost in Temporal Module, and also focus more on modeling the target appearance and relation between the template and search, we introduce one track token $\boldsymbol{F}_t^0 \in \mathbb{R}^{1 \times D}$ for each pair of images, where t means at t frame. The remaining operation in the backbone can be summarized as the following formula:

$$\begin{aligned} \boldsymbol{F}_{tzx}^0 &= Concat(\boldsymbol{F}_t^0, \boldsymbol{F}_z^0, \boldsymbol{F}_x^0), \\ \boldsymbol{F}_{tzx}^n &= Backbone(\boldsymbol{F}_{tzx}^{n-1}), n = 1...N, \end{aligned} \tag{1}$$

where N is the number layer of global attention in the backbone. Refer to Fast-iTPN(Tian et al. 2024) for more details.

Temporal Information Learning

To demonstrate the superiority of our method, we develop **three variants** of the Temporal Module. Each variant is composed of two layers, i.e., *Mamba_Cross*, *Self_Cross*, and *Self_Self*. All of them outperform most trackers, the results are shown in table 5. The outstanding performance of the above variants demonstrates that our method can effectively associate contextual information through track tokens.

The input of the Temporal Module is a set of historical track token $\mathbf{T}$ containing the appearance information of the target at various times:

$$\mathbf{T} = Concat(\boldsymbol{F}_{t-m+1}^N, ..., \boldsymbol{F}_{t-1}^N, \boldsymbol{F}_t^N), \tag{2}$$

where m is the size of the sliding window. In Temporal Module, the track token $\boldsymbol{F}_t$ interacts with other track tokens within a sliding window.

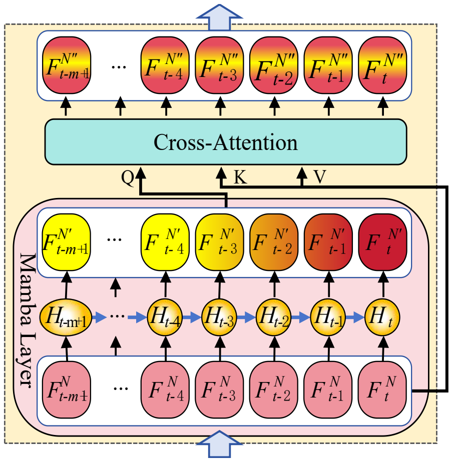


Figure 3: The schematic diagram of the Mamba_Cross. The $\boldsymbol{F}_t^N$ and $\boldsymbol{H}_t$ indicate track token and the hidden state at the t frame. After this module, the track token $\boldsymbol{F}_t^{N''}$ gathers the appearance of the previous frames within a sliding window.

Mamba_Cross. To better dig the historical target state implied in track token, we combined the mamba(Gu and Dao 2023) with long-sequences and autoregressive characteristics(Yu and Wang 2024), which demonstrate outstanding performance in long sequence tasks. We use mamba in Temporal Module. As illuminated in fig. 3, according to the principle and autoregressive characteristics of mamba, the prediction of track token $\boldsymbol{F}_t^{N'}$ depends on the previously hidden state space $\boldsymbol{H}_{t-1}$ and the current track token $\boldsymbol{F}_t^N$. After mamba, $\mathbf{T}^{'}$ is fed into a cross-attention layer and interacts with original track tokens $\mathbf{T}$. The process in the Mamba_Cross can be described as:

$$\begin{aligned} &\mathbf{T}^{'} = Mamba(\mathbf{T}),\\ &\mathbf{T}^{''} = Cross\_Attn(\mathbf{Q}=\mathbf{T}^{'}, \mathbf{K}=\mathbf{T}, \mathbf{V}=\mathbf{T}), \end{aligned} \tag{3}$$

where Q is the query, K is the key, and V is the value which is the same in following eq. (4) and eq. (5). Ultimately, $\boldsymbol{F}_t^{N''}$ gathers the target's historical appearance changes and the motion trend. Thanks to mamba's excellent ability to model sequence, in our experiment, the Temporal Module variant with mamba achieves the best performance among the three variants, i.e., 74.9% of AO in GOT-10k and 72.0% of AUC on LaSOT(Fan et al. 2019), as shown in table 5.

Self_Cross. This variant consists of a self-attention layer and a cross-attention layer. The operation in the Self_Cross can be described as the following equation:

$$\begin{aligned} &\boldsymbol{T}^{'} = Self\_Attn(\boldsymbol{Q}=\boldsymbol{T}, \boldsymbol{K}=\boldsymbol{T}, \boldsymbol{V}=\boldsymbol{T}),\\ &\boldsymbol{T}^{''} = Cross\_Attn(\boldsymbol{Q}=\boldsymbol{T}^{'}, \boldsymbol{K}=\boldsymbol{T}, \boldsymbol{V}=\boldsymbol{T}), \end{aligned} \tag{4}$$

Self_Self. In addition, we have also developed a variant that fully utilizes self-attention, namely *Self_Self*, whose operations can be expressed as the following formula:

$$\begin{aligned} &\boldsymbol{T}^{'} = Self\_Attn(\boldsymbol{Q}=\boldsymbol{T}, \boldsymbol{K}=\boldsymbol{T}, \boldsymbol{V}=\boldsymbol{T}),\\ &\boldsymbol{T}^{''} = Self\_Attn(\boldsymbol{Q}=\boldsymbol{T}^{'}, \boldsymbol{K}=\boldsymbol{T}^{'}, \boldsymbol{V}=\boldsymbol{T}^{'}). \end{aligned} \tag{5}$$

Guidance, Head and Loss

Guidance. After the Temporal Module, the T_t merging the historical appearance of the target will guide the search feature to adjust. Inspired by STARK(Yan et al. 2021), we calculate the similarity $\boldsymbol{S} \in \mathbb{R}^{N_x \times 1}$ between search spatial features $\boldsymbol{F}_x^N \in \mathbb{R}^{N_x \times D}$ and track token $\boldsymbol{F}_t^{N''} \in \mathbb{R}^{1 \times D}$. The higher the score, the greater the likelihood of the target being located. Then use an *element-wise product* to enhance the expression of the search feature.

Head and Loss. Following the popular trackers(Ye et al. 2022), we use the center-based head to predict the tracking box, which includes the position and the scale. The center-based head includes two branches, namely classification and regression. We use focal loss(Lin et al. 2017) for classification and combine GIoU loss(Rezatofighi et al. 2019) and $L1$ loss for regression. The total loss $\mathcal{L}$ is calculated as eq. (6), which $\lambda_{giou} = 2$ and $\lambda_{\mathcal{L}1} = 5$.

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_{giou}\mathcal{L}_{giou} + \lambda_{\mathcal{L}_1}\mathcal{L}_1. \tag{6}$$

Experiments

In this section, we introduce the implementation details. Then, we compare our TemTrack with SOTA methods on multiple benchmarks. Finally, we show the ablation studies to evaluate the efficiency of the proposed methods. Some tracking results and visualizations are provided to understand how TemTrack works.

Implementation Details

Our tracker is implemented in Python 3.8 using PyTorch 1.13.1. The training is on 4 NVIDIA A10 GPUs and the speed evaluation is on a single NVIDIA V100 GPU. We present two variants of TemTrack with different settings:

- **TemTrack-256.** The resolution of template image and search region is 128×128 and 256×256 pixels.
- **TemTrack-384.** The resolution of template image and search region is 192×192 and 384×384 pixels.

The Fast-iTPN(Tian et al. 2024) is used as the backbone for feature extraction and fusion, and the checkpoint of Fast-iTPN-B-224 is loaded to initialize the backbone.

Training. Following the mainstream trackers, we use four datasets for training, including COCO(Lin et al. 2014), LaSOT(Fan et al. 2019), TrackingNet(Müller et al. 2018), and GOT-10k(Huang, Zhao, and Huang 2021). Common data augmentations are used including bright jittering and horizontal flip. We train TemTrack with AdamW optimizer(Loshchilov and Hutter 2019). The learning rate of the backbone is 4×10^{-5}, and the learning rate of other parameters is 4×10^{-4}, and the weight decay is 10^{-4}. The above settings are the same as OSTrack(Ye et al. 2022). Following (Xie et al. 2024) and (Shi et al. 2024), we sample n video clips for each GPU, which contain m images as search images (all of them with the same template). So each GPU holds $n{*}m$ image pairs, i.e., the batch size is $n{*}m$. We keep the batch size equal to 32. For four GPUs, the total batch size is 128. Obviously, m is the size of the sliding window and the length of temporal information. In TemTrack, n and m

Model	Params	FLOPs	Speed
SeqTrack-B256(Chen et al. 2023)	89M	65G	40fps
SeqTrack-B384(Chen et al. 2023)	89M	148G	15fps
TemTrack-256(ours)	70M	24.8G	46fps
TemTrack-384(ours)	70M	55.7G	36fps

Table 1: Comparison of model Params, FLOPs, and Speed on NVIDIA V100.

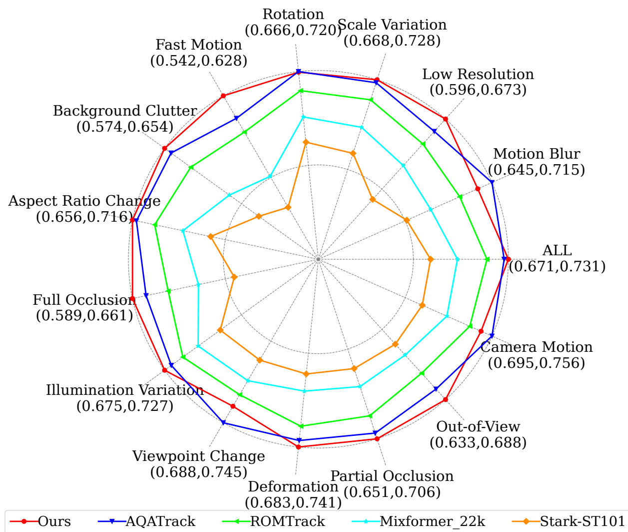
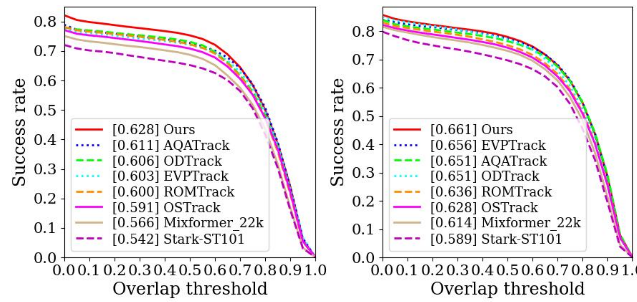


Figure 4: AUC scores of difference attributes on LaSOT(Fan et al. 2019). Best viewed in color.



(a) Fast Motion. (b) Full Occlusion.

Figure 5: Success plots of one-pass evaluation (OPE) about (a) fast motion and (b) full occlusion challenges on LaSOT. Best viewed in color and zooming in.

are 4 and 8, respectively. We train the TemTrack with 150 epochs and $60k$ image pairs for each epoch. We decrease the learning rate by the factor of 10 after the $120th$ epoch. For the GOT-10k benchmark, we train the model with only 40 epochs and the learning rate decays at 80% epochs.

Inference. During inference, the track token gather the temporal information via the Temporal Module within a sliding window. After that, the track token that contains historical appearance and motion trend conducts the search feature to adjust. Following the mainstream tracker(Chen et al. 2021; Ye et al. 2022; Xie et al. 2024; Shi et al. 2024), we utilize the Hamming window to introduce the positional priors. Also, we present the Params, FlOPs, and speed of TemTrack in the table 1. Our TemTrack-384 with very less FLOPs runs in real-time at 36 fps, faster twice than SeqTrack(Chen et al. 2023) that introduces temporal information by inputting more templates.

Results and Comparisons

We compare our evaluation results with other SOTA methods on six benchmarks to prove our effectiveness.

LaSOT(Fan et al. 2019). LaSOT (Fan et al. 2019) is a high-quality benchmark for long-term challenge on single object tracking. It consists of 1120 sequences for training and 280 sequences for testing. To show the robustness of our tracker, we compare our tracker with many SOTA trackers in fig. 1. Benefiting from the track token and Temporal Module, TemTrack learn the appearance changes and motion trends well. TemTrack achieves a new state-of-art result. As shown in table 2, TemTrack-256 obtain 72.0% of AUC, which outperforms AQATrack by 0.6%. We compare TemTrack-384 with four famous trackers in different challenges of LaSOT in fig. 4. TemTrack outperforms others in many challenges, such as fast motion, low resolution, and full occlusion. As illuminated in fig. 5, TemTrack significantly outperforms other trackers when encountering fast motion and full occlusion, outperforming ODTrack by 2.2% and 1.0% of success rate. The above outstanding performances on this long-term benchmark show the effectiveness of TemTrack in temporal information learning.

LaSOT$_{ext}$(Fan et al. 2021). This benchmark is an expansion of LaSOT(Fan et al. 2019) with additional 150 longtem sequences, introducing many challenges, such as fast-moving small objects. In table 2, we show the result of TemTrack that indicate our TemTrack outperform other trackers by a substantial margin, obtaining the highest AUC, P$_{nrom}$, and P. TemTrack achieves 52.4% of AUC, outperforming 1.2% than AQATrack(Xie et al. 2024). The excellent performances show our tracker not only mines temporal information but also addresses fast-moving small objects well.

TrackingNet(Müller et al. 2018). TrackingNet is a large-scale tracking dataset with more than 30,000 sequences for training and 511 sequences for testing. This benchmark focuses on some challenges when tracking objects in the wild, such as background clutter, full occlusion, and low resolution. We show the result of TemTrack and some SOTA trackers on TrackingNet(Müller et al. 2018) in table 3. Our tracker achieves the 85.0% of AUC score which demonstrates the robustness of TemTrack in the field.

GOT-10k(Huang, Zhao, and Huang 2021). GOT-10k is a large high-diversity benchmark for generic object tracking, which introduces a one-shot protocol for evaluation, i.e., the training and test classes are zero-overlapped. Adhering to this protocol to train our tracker, we evaluate the tracker on GOT-10k to demonstrate our generalization. As shown in

Method	Source	LaSOT			LaSOT_{ext}			GOT-10k*			TrackingNet		
		AUC	P_{norm}	P	AUC	P_{norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$	AUC	P_{norm}	P
TemTrack-256	Ours	**72.0**	**82.1**	**79.1**	**52.4**	**63.3**	**60.2**	**74.9**	**84.8**	71.7	**84.3**	**88.8**	**83.5**
AQATrack-256(Xie et al. 2024)	CVPR24	71.4	81.9	78.6	51.2	62.2	58.9	73.8	83.2	**72.1**	83.8	88.6	83.1
EVPTrack-224(Shi et al. 2024)	AAAI24	70.4	80.9	77.2	48.7	59.5	55.1	73.3	83.6	70.7	83.5	88.3	-
F-BDMTrack-256(Yang et al. 2023)	ICCV23	69.9	79.4	75.8	47.9	57.9	54.0	72.7	82.0	69.9	83.7	88.3	82.6
ROMTrack-256(Cai et al. 2023)	ICCV23	69.3	78.8	75.6	48.9	59.3	55.0	72.9	82.9	70.2	83.6	88.4	82.7
ARTrack-256(Wei et al. 2023)	CVPR23	70.4	79.5	76.6	46.4	56.5	52.3	73.5	82.2	70.9	84.2	88.7	**83.5**
SeqTrack-B256(Chen et al. 2023)	CVPR23	69.9	79.7	76.3	49.5	60.8	56.3	74.7	84.7	71.8	83.3	88.3	82.2
VideoTrack(Xie et al. 2023)	CVPR23	70.2	-	76.4	-	-	-	72.9	81.9	69.8	83.8	88.7	83.1
MixFormer-22k(Cui et al. 2022)	CVPR22	69.2	78.7	74.7	-	-	-	70.7	80.0	67.8	83.1	88.1	81.6
OSTrack-256(Ye et al. 2022)	ECCV22	69.1	78.7	75.2	47.4	57.3	53.3	71.0	80.4	68.2	83.1	87.8	82.0
STARK-ST101(Yan et al. 2021)	ICCV21	67.1	77.0	-	-	-	-	68.8	78.1	64.1	82.0	86.9	-
TransT (Chen et al. 2021)	CVPR21	64.9	73.8	69.0	-	-	-	67.1	76.8	60.9	81.4	86.7	80.3
Ocean (Zhang et al. 2020)	ECCV 20	56.0	65.1	56.6	-	-	-	61.1	72.1	47.3	-	-	-
SiamRPN++(Li et al. 2019)	CVPR19	49.6	56.9	49.1	34.0	41.6	39.6	51.7	61.6	32.5	73.3	80.0	69.4
ECO (Danelljan et al. 2017)	ICCV 17	32.4	33.8	30.1	22.0	25.2	24.0	31.6	30.9	11.1	-	-	-
SiamFC (Bertinetto et al. 2016)	ECCVW16	33.6	42.0	33.9	23.0	31.1	26.9	34.8	35.3	9.8	-	-	-
Some Trackers with Higher Resolution													
OSTrack-384(Ye et al. 2022)	ECCV22	71.1	81.1	77.6	50.5	61.3	57.6	73.7	83.2	70.8	83.9	88.5	83.2
ROMTrack-384(Cai et al. 2023)	ICCV23	71.4	81.4	78.2	51.3	62.4	58.6	74.2	84.3	72.4	84.1	89.0	83.7
F-BDMTrack-384(Yang et al. 2023)	ICCV23	72.0	81.5	77.7	50.8	61.3	57.8	75.4	84.3	72.9	84.5	89.0	84.0
SeqTrack-B384(Chen et al. 2023)	CVPR23	71.5	81.1	77.8	50.5	61.6	57.5	74.5	84.3	71.4	83.9	88.8	83.6
ARTrack-384(Wei et al. 2023)	CVPR23	72.6	81.7	79.1	51.9	62.0	58.5	75.5	84.3	74.3	**85.1**	89.1	**84.8**
HIPTrack(Cai, Liu, and Wang 2024)	CVPR24	72.7	82.9	79.5	53.0	64.3	60.6	**77.4**	**88.0**	74.5	84.5	89.1	83.8
AQATrack-384(Xie et al. 2024)	CVPR24	72.7	82.9	80.2	52.7	64.2	60.8	76.0	85.2	**74.9**	84.8	**89.3**	84.3
TemTrack-384	Ours	**73.1**	**83.0**	**80.7**	**53.4**	**64.8**	**61.0**	76.1	84.9	74.4	85.0	**89.3**	**84.8**

Table 2: Performance comparisons with state-of-the-art trackers on the test set of LaSOT(Fan et al. 2019), $\text{LaSOT}_{\text{ext}}$(Fan et al. 2021) , GOT-10k(Huang, Zhao, and Huang 2021) and Trackingnet(Müller et al. 2018). 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 using **bold** and underlined fonts respectively.

	SiamFC	ECO	SiamRPN++	TransT	OSTrack	SeqTrack	ARTrack	F-BDMTrack	EVPTrack	AQATrack	TemTrack
UAV123	46.8	53.5	61.0	69.1	68.3	69.2	67.7	69.0	70.2	70.7	**70.8**
TNL2K	29.5	32.6	41.3	50.7	54.3	54.9	57.5	56.4	57.5	57.8	**58.8**

Table 3: Performance comparisons with state-of-the-art trackers on the TNL2K(Wang et al. 2021). The top two results are highlighted with **blod** and underlined fonts respectively.

table 2, our TemTrack achieves a competitive performance among state-of-art trackers. The high performance on this one-shot tracking benchmark demonstrates the strong discriminative ability of TemTrack for unseen classes.

UAV123(Mueller, Smith, and Ghanem 2016) and TNL2K(Wang et al. 2021). We also evaluate our tracker on two additional benchmarks: UAV123 and TNL2K. They include 123 and 700 videos for testing, respectively. As shown in table 3, our TemTrack with the lower resolution of search image achieves 70.8% of AUC on UAV123 and 58.8% of AUC on TNL2K, which are better than others.

Ablation Study and Analysis

To demonstrate the effectiveness of our proposed method, we design ablation experiments from four aspects, namely ablation of TemTrack, different backbone, different components of Temporal Module, and the size of the sliding window. All the ablation study is based TemTrack-256.

Ablation Studies of TemTrack. We explore the impact of each component used in TemTrack on LaSOT(Fan et al. 2019) and GOT-10k(Huang, Zhao, and Huang 2021), as shown in table 4. The baseline based Fast-iTPN(Tian et al. 2024) consists of a backbone and a head network. For the sake of fairness, we keep the same config as the baseline for the following experiments. Firstly, we show the impact of tokens' guidance in the absence of temporal information, which outperforms 0.3% of AUC on LaSOT and outperforms 0.7% of AO on GOT-10k than the baseline. These results show that the track token can learn the target's appearance during interaction in the backbone, and help improve the expressive ability of search features. Then, we introduce

Method	LaSOT		GOT-10k	
	AUC	P_{norm}	AO	$SR_{0.5}$
Baseline	71.1	81.2	73.0	82.8
+track token	71.4	81.5	73.7	83.3
+Temporal Module	**72.0**	**82.1**	**74.9**	**84.8**

Table 4: Ablation studies of TemTrack on different dataset.

Component	AO	$SR_{0.5}$	$SR_{0.75}$
Baseline	73.0	82.8	71.6
Mamba_Cross	**74.9**	**84.8**	**71.7**
Self_Cross	74.3	84.6	71.7
Self_Self	74.2	84.2	70.8

Table 5: Influence of different layers on GOT-10k.

Backbone	Our_method	AUC	P_{norm}	P
ViT-B	-	68.6	78.4	74.3
	✓	69.6(+1.0)	79.7(+1.3)	75.5(+1.2)
HiViT	-	70.2	80.3	76.9
	✓	70.8(+0.6)	80.9(+0.6)	77.8(+0.9)
Fast-iTPN	-	71.1	81.2	78.2
	✓	**72.0**(+0.9)	**82.1**(+0.9)	**79.1**(+0.9)

Table 6: Influence of the backbone on LaSOT.

the temporal information extracted by the Temporal Module. The results show that temporal information improves the model's discriminative ability, which achieves 72.0% of AUC on LaSOT and outperforms 1.9% on GOT-10k(Huang, Zhao, and Huang 2021).

Variants of Temporal Module. Our Temporal Module consists of two layers, the first layer is mamba, and the second layer is cross-attention. To demonstrate the effectiveness of the proposed temporal information learning method, we conduct experiments using different technical approaches. Firstly, we demonstrate that using self-attention can also achieve good performance, achieving 74.3% of AO in GOT-10k, which outperforms most trackers, as demonstrated in table 5. Additionally, we demonstrate the performance when both two layers are implemented using self-attention. The results are shown in the last row of table 5, achieving 74.2% of AO in GOT-10k. Although these variants all get a comparative result, the variant that introduces the mamba with autoregressive characteristic achieves the best performance.

Different Sizes of the Sliding Window. The size of the window indicates the length of temporal information. To explore the model's potential for mining temporal information, we design different sliding window sizes m, as illuminated in table 7. When the window size is 2, the model learns a short temporal information, leading to a lower performance. When we set m to 4, the model achieves 71.9% of AUC on LaSOT(Fan et al. 2019). When we set m to 8, the model achieves 72.0% of AUC. Therefore, the optimal window size may be between 4 and 8.

Different Backbone. We prove the effect of our method by replacing different backbones, such as ViT(Dosovitskiy et al. 2021) in many trackers(Ye et al. 2022; Chen et al. 2022) and HiViT(Zhang et al. 2023) used in some recent trackers(Shi et al. 2024; Xie et al. 2024). As shown in table 6, our method based on ViT achieves 69.6% of AUC, improving by 1.0%. Otherwise, our method used HiViT as backbone achieves 70.8% of AUC, improving by 0.6%. Our method based on Fast-iTPN improves the AUC by 0.9%. The above results show the effectiveness of our method.

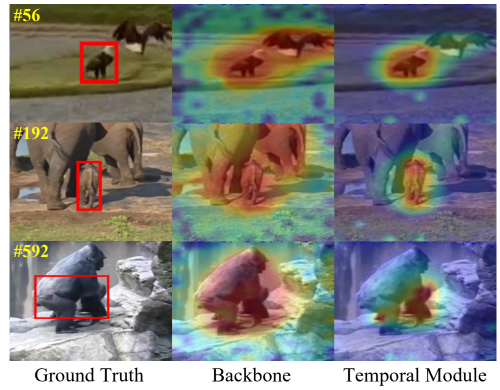


Figure 6: Visualize the attention of search to track token. The first column is ground truth, the second column is the attention in the last layer of the backbone, and the third column is the attention in Temporal Module.

m	n	AUC	P_{norm}	P
2	16	71.1	81.2	78.2
4	8	71.9	82.0	79.0
8	4	**72.0**	**82.1**	**79.1**

Table 7: Influence of window size on LaSOT.

Visualization and Qualitative Comparison. Due to the backbone focusing more on appearance modeling and introducing temporal information, TemTrack achieves the most accurate tracking in the above challenging scenes. We visualize the attention of search to the track token in backbone and Temporal Module, as demonstrated in fig. 6. In the third column, the search feature after guiding by track token indicates a more accurate location of the target in similar object interference (first row) and occlusion (last row) cases.

Conclusion

We propose a novel tracker that elegantly extracts temporal information from a list of track tokens rather than several images, reducing the model's learning and computational burden. The model's backbone focuses more on appearance modeling. Under the guidance of track token contained temporal information, the appearance features adjust to obtain more accurate tracking results. Extensive experiments on six datasets demonstrate the superiority of our method.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.U23A20383, 62472109, and 62466051), the Project of Guangxi Science and Technology (No.2024GXNSFGA010001 and 2022GXNSFDA035079), the Guangxi "Young Bagui Scholar" Teams for Innovation and Research Project, the Research Project of Guangxi Normal University (No.2024DF001).

References

Bai, Y.; Zhao, Z.; Gong, Y.; and Wei, X. 2024. ARTrackV2: Prompting Autoregressive Tracker Where to Look and How to Describe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. S. 2016. Fully-Convolutional Siamese Networks for Object Tracking. In *ECCV Workshops*, 850–865.

Cai, W.; Liu, Q.; and Wang, Y. 2024. HIPTrack: Visual Tracking with Historical Prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Cai, Y.; Liu, J.; Tang, J.; and Wu, G. 2023. Robust Object Modeling for Visual Tracking. *CoRR*, abs/2308.05140.

Cao, Z.; Huang, Z.; Pan, L.; Zhang, S.; Liu, Z.; and Fu, C. 2023. Towards real-world visual tracking with temporal contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen, B.; Li, P.; Bai, L.; Qiao, L.; Shen, Q.; Li, B.; Gan, W.; Wu, W.; and Ouyang, W. 2022. Backbone is All Your Need: A Simplified Architecture for Visual Object Tracking. In *ECCV (22)*, 375–392.

Chen, X.; Peng, H.; Wang, D.; Lu, H.; and Hu, H. 2023. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14572–14581.

Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; and Lu, H. 2021. Transformer Tracking. In *CVPR*, 8126–8135.

Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; and Ji, R. 2020. Siamese Box Adaptive Network for Visual Tracking. In *CVPR*, 6667–6676.

Cheng, L.; Wang, J.; and Li, Y. 2022. ViTrack: Efficient Tracking on the Edge for Commodity Video Surveillance Systems. *IEEE Transactions on Parallel and Distributed Systems*, 33(3): 723–735.

Cui, Y.; Jiang, C.; Wang, L.; and Wu, G. 2022. MixFormer: End-to-End Tracking with Iterative Mixed Attention. In *CVPR*, 13598–13608.

Cui, Y.; Jiang, C.; Wu, G.; and Wang, L. 2024. MixFormer: End-to-End Tracking With Iterative Mixed Attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6): 4129–4146.

Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2017. ECO: Efficient Convolution Operators for Tracking. In *CVPR*, 6931–6939.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Fan, H.; Bai, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Harshit; Huang, M.; Liu, J.; Xu, Y.; Liao, C.; Yuan, L.; and Ling, H. 2021. LaSOT: A High-quality Large-scale Single Object Tracking Benchmark. *Int. J. Comput. Vis.*, 439–461.

Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; and Ling, H. 2019. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In *CVPR*, 5374–5383.

Fu, Z.; Fu, Z.; Liu, Q.; Cai, W.; and Wang, Y. 2022. Sparsett: Visual tracking with sparse transformers. *arXiv preprint arXiv:2205.03776*.

Fu, Z.; Liu, Q.; Fu, Z.; and Wang, Y. 2021. STMTrack: Template-Free Visual Tracking With Space-Time Memory Networks. In *CVPR*, 13774–13783.

Gao, S.; Zhou, C.; Ma, C.; Wang, X.; and Yuan, J. 2022. AiATrack: Attention in Attention for Transformer Visual Tracking. In *ECCV (22)*, 146–164.

Gao, S.; Zhou, C.; and Zhang, J. 2023. Generalized Relation Modeling for Transformer Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18686–18695.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, X.; Zhong, B.; Liang, Q.; Zhang, S.; Li, N.; and Li, X. 2024a. Towards Modalities Correlation for RGB-T Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*.

Hu, X.; Zhong, B.; Liang, Q.; Zhang, S.; Li, N.; Li, X.; and Ji, R. 2024b. Transformer Tracking via Frequency Fusion. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(2): 1020–1031.

Huang, L.; Zhao, X.; and Huang, K. 2021. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5): 1562–1577.

Huang, T.; Pei, X.; You, S.; Wang, F.; Qian, C.; and Xu, C. 2024. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 1106–1114.

Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; and Yan, J. 2019. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In *CVPR*, 4282–4291.

Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; and Ling, H. 2022. Swintrack: A simple and strong baseline for transformer tracking. *Advances in Neural Information Processing Systems*, 35: 16743–16754.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *ICCV*, 2999–3007.

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, 740–755.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024. VMamba: Visual State Space Model. *arXiv preprint arXiv:2401.10166*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*, 9992–10002. IEEE.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

Ma, J.; Li, F.; and Wang, B. 2024. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*.

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

Müller, M.; Bibi, A.; Giancola, S.; Al-Subaihi, S.; and Ghanem, B. 2018. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In *ECCV*, 310–327.

Pereira, R.; Carvalho, G.; Garrote, L.; and Nunes, U. J. 2022. Sort and deep-SORT based multi-object tracking for mobile robotics: Evaluation with new data association metrics. *Applied Sciences*, 12(3): 1319.

Premachandra, C.; Ueda, S.; and Suzuki, Y. 2020. 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.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I. D.; and Savarese, S. 2019. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In *CVPR*, 658–666.

Shehzed, A.; Jalal, A.; and Kim, K. 2019. 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)*, 163–168.

Shi, L.; Zhong, B.; Liang, Q.; Li, N.; Zhang, S.; and Li, X. 2024. Explicit Visual Prompts for Visual Object Tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4838–4846.

Song, Z.; Luo, R.; Yu, J.; Chen, Y.-P. P.; and Yang, W. 2023. Compact Transformer Tracker with Correlative Masked Modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Song, Z.; Yu, J.; Chen, Y.-P. P.; and Yang, W. 2022. Transformer tracking with cyclic shifting window attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8791–8800.

Tian, Y.; Xie, L.; Qiu, J.; Jiao, J.; Wang, Y.; Tian, Q.; and Ye, Q. 2024. Fast-iTPN: Integrally Pre-Trained Transformer Pyramid Network with Token Migration. arXiv:2211.12735.

Wang, X.; Shu, X.; Zhang, Z.; Jiang, B.; Wang, Y.; Tian, Y.; and Wu, F. 2021. Towards More Flexible and Accurate Object Tracking With Natural Language: Algorithms and Benchmark. In *CVPR*, 13763–13773.

Wei, X.; Bai, Y.; Zheng, Y.; Shi, D.; and Gong, Y. 2023. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9697–9706.

Xie, F.; Chu, L.; Li, J.; Lu, Y.; and Ma, C. 2023. VideoTrack: Learning to Track Objects via Video Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22826–22835.

Xie, J.; Zhong, B.; Mo, Z.; Zhang, S.; Shi, L.; Song, S.; and Ji, R. 2024. Autoregressive Queries for Adaptive Tracking with Spatio-Temporal Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19300–19309.

Xing, Z.; Ye, T.; Yang, Y.; Liu, G.; and Zhu, L. 2024. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation. *arXiv preprint arXiv:2401.13560*.

Xu, Y.; Wang, Z.; Li, Z.; Ye, Y.; and Yu, G. 2020. SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines. In *AAAI*, 12549–12556.

Xue, C.; Zhong, B.; Liang, Q.; Xia, H.; and Song, S. 2024. Unifying Motion and Appearance Cues for Visual Tracking via Shared Queries. *IEEE Transactions on Circuits and Systems for Video Technology*.

Yan, B.; Peng, H.; Fu, J.; Wang, D.; and Lu, H. 2021. Learning Spatio-Temporal Transformer for Visual Tracking. In *ICCV*, 10428–10437.

Yang, D.; He, J.; Ma, Y.; Yu, Q.; and Zhang, T. 2023. Foreground-Background Distribution Modeling Transformer for Visual Object Tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10117–10127.

Ye, B.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2022. Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework. In *ECCV (22)*, 341–357.

Yu, W.; and Wang, X. 2024. MambaOut: Do We Really Need Mamba for Vision? *arXiv preprint arXiv:2405.07992*.

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

Zhang, X.; Tian, Y.; Xie, L.; Huang, W.; Dai, Q.; Ye, Q.; and Tian, Q. 2023. HiViT: A Simpler and More Efficient Design of Hierarchical Vision Transformer. In *International Conference on Learning Representations*.

Zhang, Z.; Peng, H.; Fu, J.; Li, B.; and Hu, W. 2020. Ocean: Object-Aware Anchor-Free Tracking. In *ECCV*, 771–787.

Zheng, Y.; Zhong, B.; Liang, Q.; Mo, Z.; Zhang, S.; and Li, X. 2024. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7588–7596.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.