Exploring Reliable Spatiotemporal Dependencies for Efficient Visual Tracking

Junze Shi[1,2,3], **Yang Yu**[1,2], **Jian Shi**[1,2,3], **Haibo Luo**[1,2*]

[1]Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences
[2]Shenyang Institute of Automation, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
shijunze@sia.cn, yuyang@sia.cn, shijian@sia.cn, luohb@sia.cn

Abstract

Recent advances in transformer-based lightweight object tracking have established new standards across benchmarks, leveraging the global receptive field and powerful feature extraction capabilities of attention mechanisms. Despite these achievements, existing methods universally employ sparse sampling during training—utilizing only one template and one search image per sequence—which fails to comprehensively explore spatiotemporal information in videos. This limitation constrains performance and causes the gap between lightweight and high-performance trackers. To bridge this divide while maintaining real-time efficiency, we propose STDTrack, a framework that pioneers the integration of reliable spatiotemporal dependencies into lightweight trackers. Our approach implements dense video sampling to maximize spatiotemporal information utilization. We introduce a temporally propagating spatiotemporal token to guide per-frame feature extraction. To ensure comprehensive target state representation, we design the Multi-frame Information Fusion Module (MFIFM), which augments current dependencies using historical context. The MFIFM operates on features stored in our constructed Spatiotemporal Token Maintainer (STM), where a quality-based update mechanism ensures information reliability. Considering the scale variation among tracking targets, we develop a multi-scale prediction head to dynamically adapt to objects of different sizes. Extensive experiments demonstrate state-of-the-art results across six benchmarks. Notably, on GOT-10k, STDTrack rivals certain high-performance non-real-time trackers ($e.g.$, MixFormer) while operating at 192 FPS (GPU) and 41 FPS (CPU).

Introduction

Visual object tracking, a cornerstone task in computer vision, aims to continuously localize arbitrary objects in video sequences based on their initial states. This technology finds widespread applications in autonomous driving systems, pedestrian detection, and unmanned aerial vehicle (UAV) operations. However, prior research has predominantly focused on enhancing tracker accuracy at the expense of operational efficiency (Wei et al. 2023; Chen et al. 2023; Cai et al. 2023; Shi et al. 2024; Xu et al. 2025), rendering most high-performance trackers impractical for deployment in resource-constrained environments. Although numerous lightweight tracking models and methodologies have been proposed (Chen et al. 2022; Blatter et al. 2021; Zaveri et al. 2025), existing efficient trackers typically exhibit significant performance degradation, resulting in a substantial performance gap compared to mainstream non-real-time counterparts.

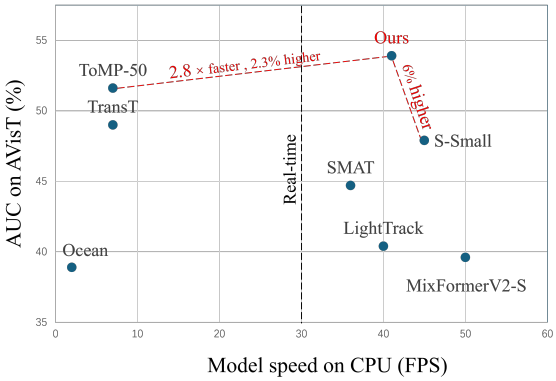

Figure 1: Comparison of our STDTrack with others on the AVisT dataset (Noman et al. 2022) on a CPU. The success score (AUC) (vertical axis) and speed (horizontal axis) are shown. Our tracker achieves substantial accuracy improvement over other state-of-the-art efficient trackers.

Early researchers attempt to develop real-time trackers predominantly focused on CNN-based architectures (Yan et al. 2021b; Borsuk et al. 2022), achieving notable computational efficiency. However, these approaches suffer from insufficient interaction between template and search regions, often leading to suboptimal tracking accuracy. To address this limitation, transformer has been introduced into lightweight trackers to enhance model performance through template-guided feature extraction (Zheng et al. 2024a; Gopal and Amer 2024). However, there is still a significant performance gap compared to state-of-the-art non-real-time trackers. In this work, we aim to bridge the performance gap between lightweight and advanced non-real-time trackers while maintaining real-time capability.

We observe that existing lightweight trackers primarily improve performance by enhancing feature extraction ca-

*Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

pabilities ($e.g.$, FERMT (Zheng et al. 2024a) boosts feature representation through optimized relational modeling in attention mechanisms). These methods typically sample only one template and one search image per sequence during training, resulting in underutilization of spatiotemporal features through sparse sampling. To address this issue, we propose STDTrack - an innovative lightweight tracking framework incorporating reliable spatiotemporal dependencies. Specifically, our method implements dense sampling of video sequences during training to enhance dataset utilization. For each sampled frame, we introduce a spatiotemporal token to encapsulate target-specific information. To fully exploit spatiotemporal correlations and enable historical state guidance for current frame tracking, we propose a Multi-frame Information Fusion Module (MFIFM).

Furthermore, to enhance the reliability of spatiotemporal dependencies, we construct a Spatiotemporal Token Maintainer (STM) that dynamically assesses the quality of all tokens and prunes the lowest-quality ones upon the introduction of new tokens. This mechanism ensures persistent guidance from historically reliable dependencies during tracking, significantly improving operational stability and robustness. Additionally, we devise a multi-scale prediction head to enhance target size adaptability of the tracker, while employing structural re-parameterization techniques (Ding et al. 2021; Chu, Li, and Zhang 2022) during inference to mitigate computational overhead caused by increased parameters.

Based on the above work, our approach achieves substantial performance improvements while maintaining real-time capability. As illustrated in Figure 1, our proposed STDTrack attains an AUC of 53.9% on the AVisT benchmark, surpassing the recent state-of-the-art tracker S-Small (Zaveri et al. 2025) by 6%. Meanwhile, STDTrack outperforms the high-performance tracker ToMP-50 (Mayer et al. 2022) while running $2.8\times$ faster.

Our main contributions are summarized as follows:

- We propose a novel lightweight tracking framework STDTrack that integrating reliable spatiotemporal dependencies during tracking. Compared to related work, for the first time, we incorporate continuous spatiotemporal information into lightweight tracking model.
- To obtain reliable spatiotemporal dependencies, we design a Multi-frame Information Fusion Module (MFIFM) to exploit contextual information across video sequences. Concurrently, a Spatiotemporal Token Maintainer (STM) is constructed to enhance the reliability of propagated dependencies.
- A multi-scale prediction head is proposed to improve the tracker's adaptability to targets of varying sizes.
- Extensive experiments demonstrate that our tracker achieves outstanding performance, attaining SOTA results among real-time trackers across multiple benchmarks. Notably, our STDTrack even surpasses the high-performance tracker MixFormer on the GOT-10k dataset.

Related Work

Visual Object Tracking based on Transformer. In the early stage of visual tracking research, practitioners predominantly employed shared-parameter Siamese networks (Bertinetto et al. 2016; Li et al. 2018) to extract target and search region features respectively, followed by cross-correlation operations for target localization. However, these methods inherently suffer from insufficient template-search interaction, fundamentally limiting their potential for high-performance tracking. Recent advancements in transformer architectures (Dosovitskiy et al. 2021; Liu et al. 2021) have catalyzed a paradigm shift, with increasing research efforts dedicated to constructing transformer-based tracking frameworks.

STARK (Yan et al. 2021a) enhances template-search interaction by integrating transformer modules after CNN-based feature extraction. SwimTrack (Lin et al. 2021) develops a SwinTransformer-based feature representation extractor and a motion-aware fusion module, explicitly incorporating motion cues during feature aggregation. ARTRack (Wei et al. 2023) adopts an encoder-decoder architecture to directly decode target location from visual features and coordinate tokens. ARTrackV2 (Bai et al. 2024) extends this through joint trajectory-appearance autoregression, simultaneously predicting target positions and reconstructing target appearance to boost performance. ODTrack (Zheng et al. 2024b) leverages densely sampled video-clip and two temporal token propagation attention mechanisms to capture spatiotemporal information. HIPTrack (Cai, Liu, and Wang 2024) maintains a memory bank updated via a FIFO strategy, enabling the model to utilize historical cues for more accurate tracking. SPMTrack (Cai, Liu, and Wang 2025) proposes TMoE, introducing dynamic expert routing for adaptive relation modeling while enabling parameter-efficient fine-tuning. Despite their remarkable accuracy, these methods predominantly employ parameter-intensive architectures requiring GPU acceleration, limiting their applicability to lightweight trackers. To address these challenges, we design a video-level tracking framework that is more suitable for lightweight applications. Our model effectively captures spatiotemporal information with negligible sacrifice in inference speed.

Efficient Tracking Network. Increasing research efforts have been directed toward developing lightweight tracking models capable of real-time operation on resource-constrained platforms. LightTrack (Yan et al. 2021b) pioneers the application of neural architecture search (NAS) for object tracking, designing a lightweight search space and a dedicated search pipeline for tracking scenarios. FEAR (Borsuk et al. 2022) proposes a lightweight tracker with a novel dual-template representation for object model adaptation. HCAT (Chen et al. 2022) constructs a feature fusion network comprising a feature sparsification module and a hierarchical cross-attention transformer, which maintains competitive performance while reducing the computational amount. MixformerV2 (Cui et al. 2023) employs knowledge distillation to yield a more unified and efficient tracker. FERMT (Zheng et al. 2024a) introduces a non-deep feature extraction strategy within the backbone net-

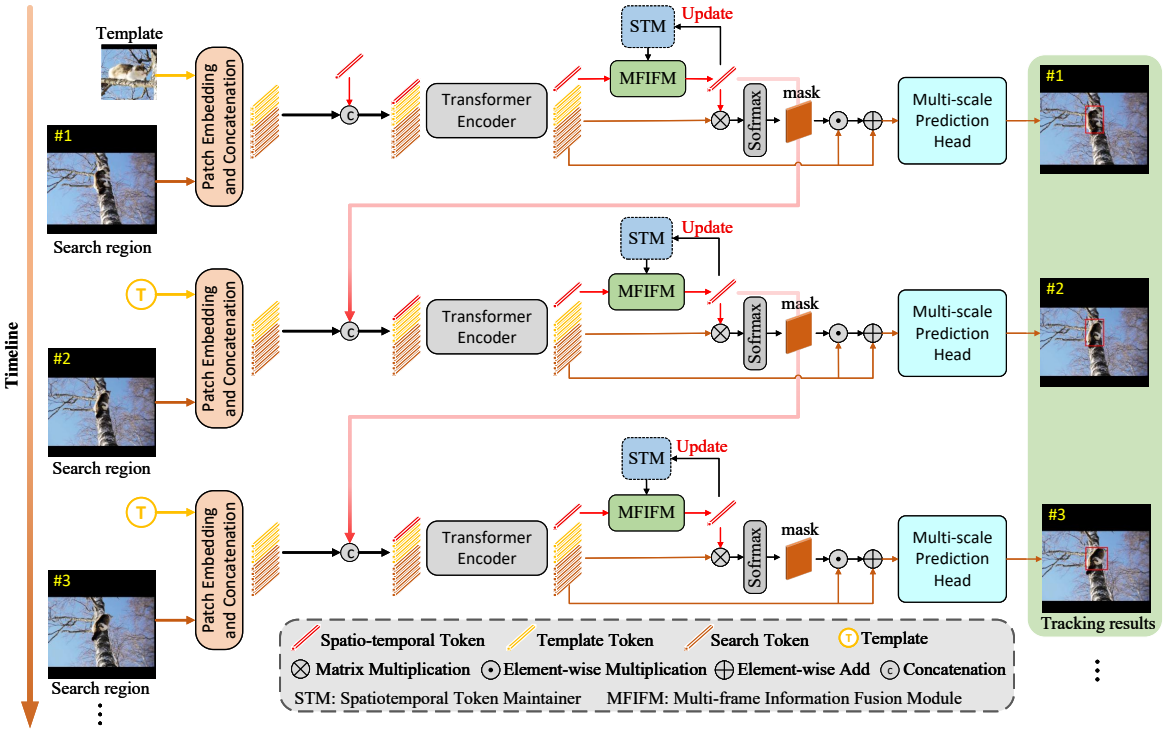

Figure 2: The architecture of our STDTrack framework. It comprises four components: a transformer encoder for feature extraction, a Multi-frame Information Fusion Module (MFIFM) that enhances and refines spatiotemporal representations, a Spatiotemporal Token Maintainer (STM) for preserving high-quality temporal dependencies and an adaptive multi-scale prediction head.

work and proposes a Dual Attention Unit (DAU) to address the performance degradation caused by model lightweighting. ORTrack (Wu et al. 2025) is designed for UAV tracking, integrating occlusion-robust representations and proposing an Adaptive Feature-Based Knowledge Distillation (AFKD) method to enhance accuracy and efficiency.

However, a performance gap persists between these lightweight trackers and high-performance counterparts. In this work, we focus on bridging this divide while maintaining real-time capability. To this end, we propose to exploit spatiotemporal features inherent in video sequences by incorporating reliable spatiotemporal dependencies during tracking, thereby significantly narrowing the accuracy gap while ensuring efficient computation.

Approach

We introduce STDTrack, a lightweight tracking framework that combines spatiotemporal dependencies, as shown in Figure 2. In this section, we elaborate on the specific components of our model: transformer encoder, Multi-frame Information Fusion Module (MFIFM), Spatiotemporal Token Maintainer (STM) and the multi-scale prediction head.

Overview

The proposed STDTrack framework uses densely sampled video sequences, taking one template and video-clip as inputs. Unlike previous lightweight trackers that employ static training paradigms, our method adopts a dynamic training strategy where historical spatiotemporal tokens from preceding timesteps are preserved and propagated temporally to assist in current-frame localization. As depicted in Figure 2, after feature extraction via the transformer encoder, spatiotemporal tokens are fed into the Multi-frame Information Fusion Module (MFIFM). This module leverages historically preserved dependencies in the Spatiotemporal Token Maintainer (STM) to refine and enhance feature representations. The augmented tokens are subsequently stored in the STM as compressed temporal context. Concurrently, we apply a mask-based search tokens enhancement mechanism guided by the spatiotemporal tokens, with the reinforced search tokens ultimately driving target localization predictions.

Transformer Encoder

The transformer encoder processes template tokens, search tokens, and spatiotemporal tokens as inputs. For each frame, we introduce a spatiotemporal token to summarize target-

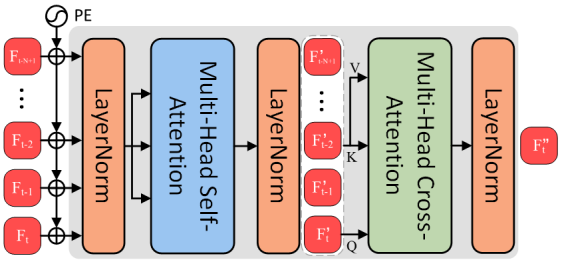

Figure 3: Architecture of MFIFM. This module fuses the spatiotemporal feature vectors $\{\mathbf{F}_t\}_{t=1}^{N}$ (summarizing target information per frame) into an augmented representation F_t'' through temporal propagation of historical dependencies.

specific information. Specifically, template and search image undergo patch embedding and linear projection to generate template tokens and search tokens, denoted as $f_z \in \mathbf{R}^{N_z \times D}$ and $f_x \in \mathbf{R}^{N_x \times D}$ respectively. Here, $N_z = H_z W_z / P^2$, $N_x = H_x W_x / P^2$ and P is the resolution of each patch. Learnable positional encoding are added to each token to provide relative positional information. Subsequently, spatiotemporal tokens, template tokens, and search tokens are concatenated and fed into the transformer encoder for representation learning. Note that for the first frame, we instantiate the spatiotemporal token as a learnable vector.

Multi-frame Information Fusion Module

During continuous tracking, historical target states capture appearance and morphological dynamics over time. By incorporating such historical context into current-frame tracking, the model gains enhanced perception of target variations, thereby improving localization accuracy. We therefore propose the Multi-frame Information Fusion Module (MFIFM), designed to endow the current spatiotemporal token with awareness of historical target states. Specifically, the MFIFM processes the spatiotemporal token F_t from the transformer encoder alongside historically preserved feature vectors $\{F_{t-1}, ..., F_{t-N+1}\}$ from the STM (detailed in next section). As illustrated in Figure 3, fixed positional encoding (Vaswani et al. 2017) are added to all tokens to preserve spatial ordering relationships. These tokens then undergo layer normalization before being processed by a multi-head self-attention layer. Leveraging the global modeling capability of attention mechanisms, each token perceives historical target states to obtain preliminarily enhanced tokens $\{\mathbf{F'}_t\}_{t=1}^{N}$. Subsequently, multi-head cross-attention facilitates interaction between the current token F_t' and $\{\mathbf{F'}_t\}_{t=1}^{N}$, yielding the final enhanced spatiotemporal token F''. The process in the MFIFM can be described as:

$$
\begin{aligned}
F_{in} &= LN(\{\mathbf{F}_t\}_{t=1}^{N} + P_{fix}) \\
\{\mathbf{F'}_t\} &= LN(MSA(Q = F_{in}, K = F_{in}, V = F_{in})) \\
F_t'' &= LN(MCA(Q = F_t', K = \{\mathbf{F'}_t\}, V = \{\mathbf{F'}_t\}))
\end{aligned} \tag{1}
$$

where P_{fix} represents the fixed positional encoding. To prevent this module from introducing excessive parameters that would compromise inference speed, we employ only a single self-attention layer and a single cross-attention layer.

Mask-based Enhancement

Before feeding the search token into the prediction head, we perform a mask-based feature enhancement operation to highlights the target regions in the search area while suppressing background interference, thereby enabling the prediction head to better perceive the target and generating more accurate tracking results (as validated in the ablation study).

As shown in Figure 2, we generate the mask using MFIFM-enhanced spatiotemporal token and apply it to the search region features. Since the enhancement process incorporates spatiotemporal tokens, the prediction results are guided by spatiotemporal information, which also helps the model condense more reliable spatiotemporal dependencies during updates. A residual connection is employed to prevent information loss.

Spatiotemporal Token Maintainer

Quality assessment and filtering of spatiotemporal information are crucial during integration. Indiscriminately introducing dependencies would allow low-quality information to cause misjudgments of the target's current state, compromising tracking performance. We therefore devise a mechanism to harvest reliable spatiotemporal dependencies while dynamically pruning inferior features generated during tracking.

We construct a Spatiotemporal Token Maintainer (STM) of length N to archive spatiotemporal tokens generated during tracking. At each timestep, the enhanced spatiotemporal token F_t'' from MFIFM and the score map produced by the prediction head are recorded in this maintainer. Since the final tracking result is determined by the maximum confidence value in the score map, the map directly reflects the quality of the current spatiotemporal token. We propose using target-background saliency as the criterion for assessing spatiotemporal feature quality. Specifically, after tracking the current frame, we compute the ratio of the maximum confidence value to the sum of all values in the score map as the quality Q:

$$
Q_t = \frac{max(score)}{\sum_{i=1}^{H}\sum_{j=1}^{W} score_{ij}} \tag{2}
$$

where the resolution of score map is (H, W). A higher Q indicates greater target saliency in the prediction map, signifying more reliable spatiotemporal dependencies. When the STM has not reaches its maximum capacity N, new tokens are stored directly. At full capacity, newly generated token is recorded while the maintainer simultaneously prunes the token with the lowest recorded Q among existing entries. This update strategy ensures our model persistent access to the N most reliable spatiotemporal dependencies during tracking.

Multi-scale Prediction Head

Contemporary trackers predominantly employ dual-branch (regression and classification) convolutional prediction

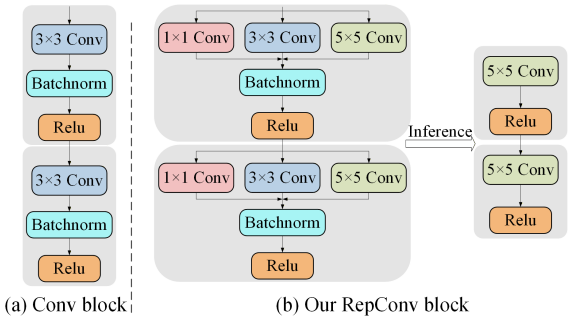

Figure 4: RepConv block design. (a) Standard Conv block in the center head (Ye et al. 2022). (b) Our proposed RepConv block employing structural re-parameterization technique. During inference, this technique merges multi-branch convolutional layers into a single convolution operation.

heads (Ye et al. 2022; Zheng et al. 2024a). The classification branch predicts target center locations, while the regression branch estimates bounding-box offsets relative to the center. We observe that homogeneous convolutional kernel sizes in these heads restrict scale adaptability, causing suboptimal performance on targets of varying dimensions ($e.g.$, small objects). To address this limitation, we propose a multi-scale prediction head that captures multi-scale representations during training, enhancing robustness to scale variations. As illustrated in Figure 4 (a), our head comprises stacked RepConv blocks. Within each block, we integrate additional 1×1 and 5×5 convolutional pathways to extract complementary features, subsequently fused through element-wise summation. The aggregated features finally used to generate a score map for classification branch, and local offsets and normalized bounding-box size for regression branch.

However, directly adding convolutional pathways would introduce excessive computational load, adversely affecting inference speed. We therefore utilize structural re-parameterization to merge multi-branch architectures into a single branch during inference. This mathematically equivalent transformation preserves model performance without compromise. For the proposed RepConv block, we first expand all convolution kernels to 5×5 dimensions through zero-padding to prepare for subsequent kernel fusion. Next, we consolidate the equivalent 5×5 convolution kernels and bias terms. Finally, we integrating convolution and batchnorm layer. This process is formally expressed as:

$$\begin{aligned}
\widehat{K}^{(1)}_{:,:,i,j} &= \begin{cases} K^{(1)} & i=2, j=2 \\ 0 & otherwise \end{cases} \\
\widehat{K}^{(3)}_{:,:,i,j} &= \begin{cases} K^{(3)}_{:,:,i,j} & 1 \le i \le 3, 1 \le j \le 3 \\ 0 & otherwise \end{cases} \\
\widehat{K}^{(5)} &= K^{(5)}
\end{aligned} \tag{3}$$

$$\begin{aligned}
K_{merged} &= \widehat{K}^{(1)} + \widehat{K}^{(3)} + \widehat{K}^{(5)} \\
b_{merged} &= \widehat{b}^{(1)} + \widehat{b}^{(3)} + \widehat{b}^{(5)}
\end{aligned} \tag{4}$$

$$\begin{aligned}
\widehat{K}_{merged} &= \frac{\gamma}{\sqrt{\sigma^2 + \epsilon}} \cdot K_{merged} \\
\widehat{b}_{merged} &= \beta + \frac{\gamma \cdot (b_{merged} - \mu)}{\sqrt{\sigma^2 + \epsilon}}
\end{aligned} \tag{5}$$

where $\widehat{K}$ denote the expanded convolution kernel. μ, σ^2, γ, β are the running mean, running variance, and learned scaling factor and bias of the BN layer. ϵ is a small positive constant.

Training Loss

The loss function combines focal loss (Lin et al. 2017) for classification with L1 loss and GIoU loss (Rezatofighi et al. 2019) for regression, formally expressed as:

$$L_{total} = \lambda_{cls} L_{cls} + \lambda_{giou} L_{giou} + \lambda_1 L_1 \tag{6}$$

where λ_{cls}, λ_{giou} and λ_1 represent the weight values of each loss function.

Experiments

Implementation Details

Our tracker is implemented using Python 3.9.7 and PyTorch 2.0.0. All models were trained on a single NVIDIA GeForce RTX 4090.

Training. During training, we implement dense sampling of video sequences to provide richer spatiotemporal information. The maximum sampling interval is set to 200, with a video clip length of 8 frames. Video clips are reversed with 0.5 probability as a data augmentation strategy to enhance model robustness. The training dataset comprises GOT-10k (Huang, Zhao, and Huang 2021), TrackingNet (Müller et al. 2018), LaSOT (Fan et al. 2018), and COCO (Lin et al. 2014). We employ ViT-tiny (Gao et al. 2024) as our transformer encoder. Search images are resized to 256×256, while template is resized to 128×128 as model inputs. We optimize model parameters using AdamW with differential learning rate: 4e-5 for the transformer encoder and 4e-4 for the rest, coupled with 1e-4 weight decay. Training proceeds for 500 epochs, with learning rate decay initiating at epoch 400. For fair evaluation on GOT-10k, we adhere to the one-shot protocol, exclusively training on GOT-10k for 100 epochs and initiating decay at epoch 80.

Inference. During inference, the model first applies structural re-parameterization to transform the prediction head into the optimized architecture shown in Figure 4 (b), enhancing computational efficiency. Spatiotemporal tokens are continuously extracted and stored in the STM. Upon reaching maximum capacity, the STM dynamically updates stored tokens based on quality assessments to ensure reliable dependencies. A Hanning window is applied to the predicted score map for motion trajectory smoothing. Our STDTrack achieves real-time performance at 41 FPS on CPU platforms and 192 FPS on GPU devices.

Comparison with the SOTA

We evaluate our STDTRack on 6 challenging benchmarks: GOT-10k (Huang, Zhao, and Huang 2021), TrackingNet (Müller et al. 2018), LaSOT (Fan et al. 2018),

Method	Source	GOT-10k*			TrackingNet			LaSOT			Speed(fps)	
		AO	$SR_{0.5}$	$SR_{0.75}$	AUC	P_{norm}	P	AUC	P_{norm}	P	GPU	CPU
CPU non-real-time Methods												
STARK-ST50	ICCV21	68.0	77.7	62.3	81.3	86.1	-	66.6	-	-	120	16
TransT	CVPR21	67.1	76.8	60.9	81.4	86.7	80.3	64.9	73.8	69.0	157	3
MixFormer-22k	CVPR22	70.7	80.0	67.8	83.1	88.1	81.6	69.2	78.7	74.7	106	8
OSTrack_{256}	CVPR22	71.0	80.4	68.2	83.1	87.8	82.0	69.1	78.7	75.2	145	14
ARTrack_{256}	CVPR23	73.5	82.2	70.9	84.2	88.7	83.5	70.4	79.5	76.6	62	5
ROMTrack_{256}	ICCV23	72.9	82.9	70.2	83.6	88.4	82.7	69.3	78.8	75.6	99	10
EVPTrack_{224}	AAAI24	73.3	83.6	70.7	83.5	88.3	-	70.4	80.9	77.2	112	12
LMTrack_{256}	AAAI25	76.3	87.1	73.9	84.2	89.0	82.8	69.8	79.2	76.3	101	8
SPMTrack-B	CVPR25	76.5	85.9	76.3	86.1	90.2	85.6	74.9	84.0	81.7	-	-
CPU real-time Methods												
LightTrack	CVPR21	61.1	71.0	-	72.5	77.8	69.5	53.8	-	53.7	91	40
STARK-Lighting	ICCV21	59.6	69.6	47.9	72.7	77.9	67.4	57.8	66.0	57.4	234	63
FEAR-XS	ECCV22	61.9	72.2	-	-	-	-	53.5	-	54.5	252	85
HCAT-B	ECCV22	65.1	76.5	56.7	76.6	82.6	72.9	59.3	68.7	61.0	371	21
MixFormerV2-S	NIPS23	-	-	-	75.8	81.1	70.4	60.6	69.9	60.4	312	50
SMAT	WACV24	64.5	74.7	57.8	78.6	84.2	75.6	61.7	71.1	64.6	168	36
FERMT	ECCV24	<u>69.6</u>	<u>80.1</u>	<u>63.2</u>	<u>80.8</u>	<u>85.9</u>	<u>78.1</u>	<u>65.1</u>	<u>74.6</u>	<u>69.1</u>	225	46
S-Small	WACV25	64.6	75.1	-	78.4	83.5	74.6	60.7	-	62.2	254	45
STDTrack	ours	**71.3**	**81.6**	**66.2**	**81.4**	**86.1**	**79.0**	**67.3**	**77.3**	**72.2**	192	41

Table 1: Comparison on three test set of GOT-10k, Trackingnet, and LaSOT. $*$ denotes models that are trained exclusively on the GOT-10k dataset. The best two results are in **bold** and <u>underline</u>, respectively.

AVisT (Noman et al. 2022), NFS(Galoogahi et al. 2017), and UAV123 (Mueller, Smith, and Ghanem 2016). To ensure fair comparison, we use the same evaluation metrics as (Zheng et al. 2024a), and all inference speed results are measured under identical hardware configurations.

GOT-10k. GOT-10k is a large-scale, generic object tracking benchmark containing over 10,000 videos with more than 1.5 million manually annotated bounding boxes. It features a diverse taxonomy of 563 object classes and emphasizes motion diversity to reflect real-world tracking challenges. As shown in Tab. 1, STDTrack achieves state-of-the-art performance among real-time trackers, surpassing the previous best method FERMT by 1.7% in AO. Notably, STDTrack outperforms the recent non-real-time tracker MixFormer while operating 1.8× faster on GPU and 5× faster on CPU platforms.

TrackingNet. TrackingNet offers an extensive dataset for large-scale training and evaluation of visual object trackers. It comprises over 30,000 videos with more than 14 million dense bounding box annotations. Its scale and diversity in object categories, scenes, and motion patterns make it suitable for assessing performance across varied real-world scenarios. As evidenced in Tab. 1, our STDTrack surpasses the real-time tracker S-Small by significant margins, outperforming 2.0%, 2.6%, and 4.4% in terms of AUC, normalized precision and precision. Simultaneously, it achieves competitive performance against non-real-time trackers ($e.g.$, STARK-ST50, TransT (Chen et al. 2021)), demonstrating exceptional generalization capability and robustness.

LaSOT. LaSOT is a high-quality benchmark specifically designed for long-term tracking evaluation. It contains 1400 sequences with an average length of over 2500 frames. As presented in Tab. 1, our tracker establishes new SOTA among real-time methods. Compared to the latest method S-Small, we achieve significant improvements of 6.6% in AUC and 10% in precision (P), thereby validating the effectiveness of spatiotemporal feature integration within our framework.

Method	Source	AVisT	NFS	UAV123
LightTrack	CVPR21	40.4	56.5	61.7
STARK-Lighting	ICCV21	-	59.6	62.0
FEAR-XS	ECCV22	38.7	48.6	61.0
HCAT-B	ECCV22	41.8	61.9	63.6
MixFormerV2-S	NIPS23	39.6	61.0	63.4
SMAT	WACV24	44.7	62.0	64.3
FERMT	ECCV24	-	<u>65.1</u>	67.5
ORTrack	CVPR25	-	-	66.4
S-Small	WACV25	<u>47.9</u>	62.4	<u>68.1</u>
STDTrack	ours	**53.9**	**65.3**	**68.4**

Table 2: Comparison of AUC metric on AVisT, NFS, and UAV123. The best two results are in **bold** and <u>underline</u>, respectively.

AVisT, NFS and UAV123. AVisT is a highly challenging benchmark comprising 120 video sequences. It involving tracking object under extreme weather conditions such as heavy rain, heavy snow, and sandstorms. NFS is a benchmark designed for evaluating trackers under fast mo-

#	Method	Params (M)	FLOPs (G)	FPS (CPU)	GOT-10k			$\Delta(\%)$
					$AO(\%)$	$SR_{0.5}(\%)$	$SR_{0.75}(\%)$	
1	Baseline	8.1	2.38	54	69.2	79.6	64.8	-
2	+ spatiotemporal token	8.1	2.41	48	69.5	79.3	65.8	+ 1.0
3	+ MFIFM	8.4	2.41	45	70.2	80.4	65.7	+ 2.7
4	+ mask-based enhancement	8.4	2.41	45	70.7	80.7	**66.5**	+ 4.3
5	+ multi-scale head	15.6	3.54	41	**71.3**	**81.6**	66.2	+ 5.5

Table 3: Ablation Study on GOT-10k. We incrementally integrating proposed modules into the baseline to quantify their contributions. Δ denotes the overall change of the three metrics against the baseline.

STM update mechanism	$AO(\%)$	$SR_{0.5}(\%)$	$SR_{0.75}(\%)$
First-in first-out	71.0	81.4	65.6
Quality-based	**71.3**	**81.6**	**66.2**

Table 4: Comparison of different STM update mechanism on GOT-10k.

#	capacity	AO(%)	$SR_{0.5}(\%)$	$SR_{0.75}(\%)$
1	2	69.6	79.3	65.4
2	4	70.9	81.2	65.9
3	6	**71.3**	**81.6**	**66.2**
4	8	69.9	79.9	65.2

Table 5: Comparison of different STM capacities on GOT-10k.

tion and motion blur challenges. This dataset includes 100 challenging videos with accurate bounding boxes, focusing on scenarios where rapid target movement is the primary difficulty. UAV123 is a benchmark captured from a low-altitude aerial perspective using UAVs, which contains 123 video sequences. These three datasets represent out-of-distribution (OOD) scenarios not encountered during training. As evidenced in Tab. 2, our STDTrack achieves SOTA performance across all benchmarks. Notably, it attains a 6.0% AUC improvement over the previous SOTA on AVisT, demonstrating exceptional generalization capability.

Ablation Study

Importance of spatiotemporal token. We first train a baseline model devoid of any proposed modules, which performs tracking using only an initial template and search image. We then introduce a spatiotemporal token to provide temporal context. The token generated in the current frame propagates to subsequent frames to guide feature extraction. From the first and second rows of Tab. 3, it can be observed that model performance improves after adding spatiotemporal token, validating the effectiveness of our token design.

Study on MFIFM. The Multi-frame Information Fusion Module (MFIFM) is one of the critical components in our model. It integrates the target's historical information into the current spatiotemporal dependencies, enabling the model to maintain a comprehensive understanding of the target's previous states during tracking. To further evaluate the effectiveness of this module, we conducted an ablation study to investigate its impact on model performance. As shown in row 2 and row 3 of Tab. 3, MFIFM achieves a substantial performance gain with only a minimal increase in parameters and computational cost. Specifically, it achieves gains of 0.7% and 1.1% on the AO and $\mathrm{SR}_{0.5}$ metrics, respectively. The results demonstrate that providing the model with richer historical information enhances its perception of the target's appearance variations throughout the tracking process, thereby contributing to more precise tracking.

Effectiveness of mask-based enhancement. To maximize the utility of the spatiotemporal tokens obtained from the current frame, we do not directly feed the output features from the transformer encoder into the prediction head. Instead, we first enhance these features using the spatiotemporal tokens, and then employ the enhanced features for target localization. As illustrated in Figure 2, we generate a mask based on the interaction between the spatiotemporal token and the search tokens. This mask serves to highlight potential target regions and suppress background distractions. The effectiveness and feasibility of this operation are validated by the results in rows 3 and 4 of Tab. 3. Specifically, incorporating our proposed mask-based enhancement further boosts model performance, yielding an additional improvement of 0.5% in the AO metric.

Multi-scale prediction head. The multi-scale prediction head enhances the model's robustness by adapting to targets of varying sizes through receptive fields at different scales. To investigate its contribution, we design experiment to assess its impact on performance. As shown in the forth and fifth rows of Tab. 3, employing the multi-scale prediction head leads to a improvement in overall performance. Despite the additional learnable parameters introduced by this module, the use of structural re-parameterization limits the inference speed penalty to a minor cost. These experimental results demonstrate its positive impact on the overall performance of the tracker.

The update mechanism of STM. To investigate the impact of different approaches to maintaining spatiotemporal dependencies, we designed two distinct STM update mechanisms. As presented in Tab. 4, we compare a First-In-First-Out (FIFO) strategy with our proposed quality-based update mechanism. Experimental results indicate that the quality-based approach achieves superior performance across all three evaluation metrics on GOT-10k. While the FIFO mechanism ensures the model consistently accesses the most

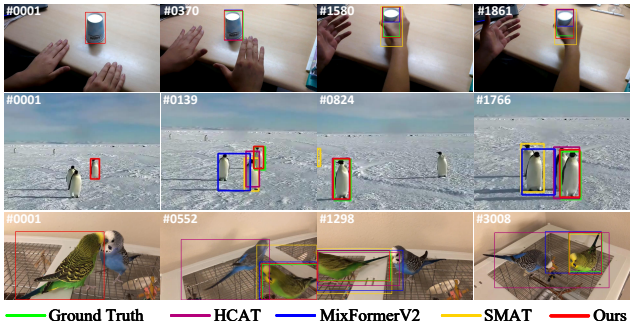

Figure 5: Qualitative comparison of our STDTrack and other SOTA trackers on LaSOT benchmark.

recent information, its lack of discriminative filtering allows low-quality spatiotemporal dependencies to mislead the model, ultimately degrading performance. In contrast, our proposed quality-based update mechanism reliably provides the model with high-fidelity spatiotemporal dependencies, effectively enhancing tracker robustness.

The capacity of STM. We design a Spatiotemporal Token Maintainer (STM) to store the target's historical state representations across past frames. It is hypothesized that larger STM capacity provides more comprehensive target context for the tracker, potentially enhancing performance. However, tracking inaccuracies in individual frames introduce discrepancies between stored states and groundtruth. These errors accumulate temporally within the maintainer. Excessively large capacity heightens the risk of storing corrupted records. We therefore conduct an ablation study on STM capacity to identify an optimum range.As shown in Tab. 5, the experimental results align with our hypothesis: tracker performance improves progressively with increasing STM capacity, but deteriorates rapidly beyond a certain threshold. For this work, we select a STM capacity of 6.

Visualization

Visualization. To demonstrate the effectiveness of our STDTrack, we compare it against three state-of-the-art lightweight trackers on the LaSOT benchmark. As shown in Figure 5, our approach achieves superior performance. This advantage stems from our integration of reliable spatiotemporal dependencies, which provide historical target states during challenging scenarios ($e.g.$, distractor interference), thereby enhancing model robustness.

Furthermore, we incrementally integrate the proposed modules into the baseline model and compare the differences in attention heatmaps generated during tracking. As shown in Figure 6, the first column marks the groundtruth position of target, while each subsequent column corresponds to the heatmap generated by the respective model listed in Tab. 3. It can be observed that after incorporating our proposed modules, the models exhibit enhanced resistance to environmental distractions compared to the baseline, allowing them to focus more accurately on the target and achieve improved tracking performance. Specifically, when the target is occluded or disturbed by similar

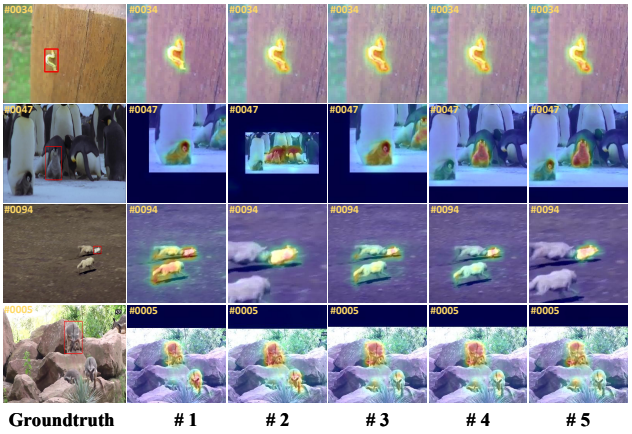

Figure 6: A comparison of heatmaps between the baseline and models with the proposed modules integrated progressively. Each column corresponds to a model in Tab. 3.

objects, the baseline model inevitably allocates attention to surrounding distractors. However, as the proposed modules are progressively added, the model gains the ability to perceive spatiotemporal information, leading to a gradual convergence of attention, which ultimately focuses entirely on the target. This observation demonstrates the effectiveness of our proposed modules and highlights the robustness and resistance to distractors of the overall approach.

Discussion and Conclusion

In this work, we present STDTrack, the first video-level framework designed for lightweight tracking. We densely sample video sequences and propagate spatiotemporal features during training, providing rich contextual information to the model. Specifically, at each timestep, we introduce a spatiotemporal token that propagates temporally to summarize target-specific information. Furthermore, we propose MFIFM to extend temporal perception to historical frames while adaptively enhancing current features. The constructed STM stores frame-level spatiotemporal tokens and employs quality-based updates to ensure dependency reliability. Additionally, our multi-scale prediction head adapts to targets of varying dimensions. These innovations collectively enable STDTrack to achieve substantial improvements while maintaining real-time capability. Extensive experiments demonstrate state-of-the-art results across six benchmarks. We anticipate this work will inspire future research in efficient visual tracking.

Limitations and future works. Our tracker may fail in certain extreme scenarios, such as when the target moves completely outside the search region or under drastic scene changes. To address this, we plan to incorporate self-assessment and global search mechanisms to enable recovery from tracking failures. Additionally, the current model does not employ compression methods like distillation or pruning. Judicious application of these techniques could make our tracker faster and more lightweight. These directions represent important avenues for our future research.

References

Bai, Y.; Zhao, Z.; Gong, Y.; and Wei, X. 2024. ARTrackV2: Prompting Autoregressive Tracker Where to Look and How to Describe. *CVPR*, 19048–19057.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. S. 2016. Fully-convolutional siamese networks for object tracking. In *Proceedings of the European conference on computer vision*, 850–865.

Blatter, P.; Kanakis, M.; Danelljan, M.; and Gool, L. V. 2021. Efficient Visual Tracking with Exemplar Transformers. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1571–1581.

Borsuk, V.; Vei, R.; Kupyn, O.; Martyniuk, T.; Krashenyi, I.; and Matas, J. 2022. FEAR: Fast, Efficient, Accurate and Robust Visual Tracker. In *European Conference on Computer Vision*.

Cai, W.; Liu, Q.; and Wang, Y. 2024. HIPTrack: Visual Tracking with Historical Prompts. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19258–19267.

Cai, W.; Liu, Q.; and Wang, Y. 2025. SPMTrack: Spatio-Temporal Parameter-Efficient Fine-Tuning with Mixture of Experts for Scalable Visual Tracking. *ArXiv*, abs/2503.18338.

Cai, Y.; Liu, J.; Tang, J.; and Wu, G. 2023. Robust Object Modeling for Visual Tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9555–9566.

Chen, X.; Peng, H.; Wang, D.; Lu, H.; and Hu, H. 2023. SeqTrack: Sequence to Sequence Learning for Visual Object Tracking. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14572–14581.

Chen, X.; Wang, D.; Li, D.; and Lu, H. 2022. Efficient Visual Tracking via Hierarchical Cross-Attention Transformer. In *ECCV*.

Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; and Lu, H. 2021. Transformer Tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8122–8131.

Chu, X.; Li, L.; and Zhang, B. 2022. Make RepVGG Greater Again: A Quantization-aware Approach. In *AAAI Conference on Artificial Intelligence*.

Cui, Y.; Cheng, J.; Wang, L.; and Wu, G. 2022. MixFormer: End-to-End Tracking with Iterative Mixed Attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13598–13608.

Cui, Y.; Song, T.; Wu, G.; and Wang, L. 2023. MixFormerV2: Efficient Fully Transformer Tracking. In *Advances in Neural Information Processing Systems*, volume 36, 58736–58751.

Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; and Sun, J. 2021. RepVGG: Making VGG-style ConvNets Great Again. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13728–13737.

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 *Int. Conf. Learn. Represent.*

Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; and Ling, H. 2018. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5369–5378.

Galoogahi, H. K.; Fagg, A.; Huang, C.; Ramanan, D.; and Lucey, S. 2017. Need for Speed: A Benchmark for Higher Frame Rate Object Tracking. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 1134–1143.

Gao, J.; Lin, S.; Wang, S.; Kou, Y.; Li, Z.; Li, L.; Zhang, C.; Zhang, X.; Wang, Y.; and Hu, W. 2024. An Experimental Study on Exploring Strong Lightweight Vision Transformers via Masked Image Modeling Pre-training. *International Journal of Computer Vision*.

Gopal, G. Y.; and Amer, M. A. 2024. Separable Self and Mixed Attention Transformers for Efficient Object Tracking. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 6694–6703.

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

Li, B.; Yan, J.; Wu, W.; Zhu, Z.; and Hu, X. 2018. High Performance Visual Tracking with Siamese Region Proposal Network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8971–8980.

Lin, L.; Fan, H.; Xu, Y.; and Ling, H. 2021. SwinTrack: A Simple and Strong Baseline for Transformer Tracking. In *Adv. Neural Inform. Process. Syst.*, 16743–16754.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999–3007.

Lin, T.-Y.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European conference on computer vision*, 740–755.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, 9992–10002.

Mayer, C.; Danelljan, M.; Bhat, G.; Paul, M.; Paudel, D. P.; Yu, F.; and Gool, L. V. 2022. Transforming Model Prediction for Tracking. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8721–8730.

Mueller, M.; Smith, N.; and Ghanem, B. 2016. A Benchmark and Simulator for UAV Tracking. In *Proceedings of the European conference on computer vision*, 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 *Proceedings of the European conference on computer vision*, 300–317.

Noman, M.; Ghallabi, W. A.; Najiha, D.; Mayer, C.; Dudhane, A.; Danelljan, M.; Cholakkal, H.; Khan, S. H.; Gool, L. V.; and Khan, F. S. 2022. AVisT: A Benchmark for Visual Object Tracking in Adverse Visibility. *ArXiv*, abs/2208.06888.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 658–666.

Shi, L.; Zhong, B.; Liang, Q.; Li, N.; Zhang, S.; and Li, X. 2024. Explicit Visual Prompts for Visual Object Tracking. In *AAAI Conference on Artificial Intelligence*.

Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Adv. Neural Inform. Process. Syst.*, 5998–6008.

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.

Wu, Y.; Wang, X.; Yang, X.; Liu, M.; Zeng, D.; Ye, H.; and Li, S. 2025. Learning Occlusion-Robust Vision Transformers for Real-Time UAV Tracking. *ArXiv*, abs/2504.09228.

Xu, C.; Zhong, B.; Liang, Q.; Zheng, Y.; Li, G.; and Song, S. 2025. Less is More: Token Context-aware Learning for Object Tracking. *AAAI*.

Yan, B.; Peng, H.; Fu, J.; Wang, D.; and Lu, H. 2021a. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10448–10457.

Yan, B.; Peng, H.; Wu, K.; Wang, D.; Fu, J.; and Lu, H. 2021b. LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15175–15184.

Ye, B.; Chang, H.; Ma, B.; and Shan, S. 2022. Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework. In *Proceedings of the European conference on computer vision*, 341–357.

Zaveri, R. J.; Patel, S.; Gu, Y.; and Doretto, G. 2025. Improving Accuracy and Generalization for Efficient Visual Tracking. *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 9468–9478.

Zheng, J.; Liang, M.; Huang, S.; and Ning, J. 2024a. Exploring the Feature Extraction and Relation Modeling For Light-Weight Transformer Tracking. In *ECCV*, 110–126.

Zheng, Y.; Zhong, B.; Liang, Q.; Mo, Z.; Zhang, S.; and Li, X. 2024b. ODTrack: Online Dense Temporal Token Learning for Visual Tracking. In *AAAI Conference on Artificial Intelligence*.