Less Is More: Token Context-Aware Learning for Object Tracking

Chenlong Xu[1], Bineng Zhong[1*], Qihua Liang[1], Yaozong Zheng[1]
Guorong Li[2], Shuxiang Song[1]
[1]Key Laboratory of Education Blockchain and Intelligent Technology, Ministry of Education,
Guangxi Normal University, Guilin 541004, China
[2]Key Laboratory of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences
xuchenlong@stu.gxnu.edu.cn, bnzhong@gxnu.edu.cn, qhliang@gxnu.edu.cn, yaozongzheng@stu.gxnu.edu.cn,
liguorong@ucas.edu.cn, songshuxiang@mailbox.gxnu.edu.cn

Abstract

Recently, several studies have shown that utilizing contextual information to perceive target states is crucial for object tracking. They typically capture context by incorporating multiple video frames. However, these naive frame-context methods fail to consider the importance of each patch within a reference frame, making them susceptible to noise and redundant tokens, which deteriorates tracking performance. To address this challenge, we propose a new token context-aware tracking pipeline named **LMTrack**, designed to automatically learn high-quality reference tokens for efficient visual tracking. Embracing the principle of ***Less** is **More***, the core idea of LMTrack is to analyze the importance distribution of all reference tokens, where important tokens are collected, continually attended to, and updated. Specifically, a novel Token Context Memory module is designed to dynamically collect high-quality spatio-temporal information of a target in an autoregressive manner, eliminating redundant background tokens from the reference frames. Furthermore, an effective Unidirectional Token Attention mechanism is designed to establish dependencies between reference tokens and search frame, enabling robust cross-frame association and target localization. Extensive experiments demonstrate the superiority of our tracker, achieving state-of-the-art results on tracking benchmarks such as GOT-10K, TrackingNet, and LaSOT.

Introduction

Object tracking is a fundamental component of computer vision, designed to localize and track an arbitrary target within a video sequence based on its initial location. To tackle this challenging task, recent research (Yan et al. 2021a; Fu et al. 2021; Cui et al. 2022; Chen et al. 2023; Zheng et al. 2024; Bai et al. 2024) constructs high-performance tracking algorithms by exploring long-term contextual relationships. Typically, researchers achieve this within multiple video frames to capture contextual information. However, these naive methods have a significant drawback: they treat frame as the smallest units of context, neglecting that the importance of each patch in a reference frame is different for target localization in the search frame. This oversight makes them susceptible to redundant noisy information of the reference frame, thereby deteriorating tracking performance.

*Corresponding author.

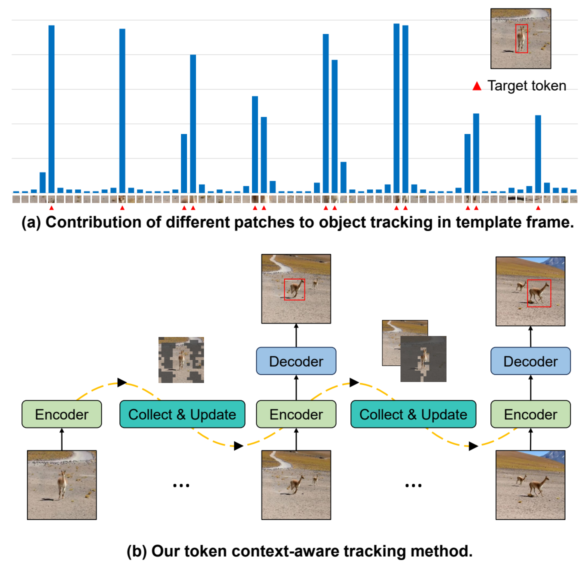

Figure 1: (a) Contribution (number of referenced) of different tokens to object tracking in a template frame. It can be observed that most background tokens are rarely referenced during the tracking process, while the target tokens retained as long-term reference cues. (b) Our token context-aware tracking method based on the token context memory module and the unidirectional token attention mechanism.

What type of reference cues play a dominant role in object tracking? To answer this question, we design a simple Transformer tracker, consisting of a Transformer network (i.e., ViT (Dosovitskiy et al. 2021)), a classification head, and a regression head, to explore the impact of each patch within the template frame on object tracking (localization) throughout the entire video sequence. During inference, we first obtain the attention map from the search frame to the template frame and the classification score map. We then perform element-wise multiplication of the two maps and average along the template dimension to calculate the importance score of each token (patch) for target localization in the search frame. As shown in Fig. 1 (a), this gives us a distribution of importance scores for all reference tokens. We observe that most background tokens are rarely referenced during the tracking process and have minimal impact on the results, while target tokens are largely retained as long-term reference cues. This validates our motivation that a few high-quality tokens play a crucial role in the tracking process.

Based on the above findings, if we continue to allocate equal attention to all reference tokens during the entire tracking process, it will increase the model's perceptual and computational load, especially when dealing with complex scenarios. Adhering to the philosophy of *less is more*, we design a simple yet effective token context-aware tracking pipeline named **LMTrack**, which automatically learns high-quality reference tokens across timestamps for visual tracking. As shown in Fig. 1 (b), the core idea of LMTrack is to analyze the importance distribution of all reference tokens, where important tokens are collected, continually attended to, and updated. Specifically, a novel Token Context Memory module is designed to dynamically collect and update high-quality spatio-temporal information of a target in an autoregressive manner, eliminating low-quality background tokens from the reference frames. This ensures that fewer reference tokens are used for accurate target localization in the search frames. Our approach discards the traditional frame-level context with redundant low-quality background information and instead uses a fine-grained token-level context to represent important reference cues across times-steps. This distinguishes our model significantly from other works (Chen et al. 2023; Zheng et al. 2024). Furthermore, an effective Unidirectional Token Attention mechanism is designed to establish dependencies between reference tokens and the search frame in a unidirectional propagation manner, enabling robust cross-frame association and target localization.

Through this new modeling approach, we delegate the decision-making for target reference information to the tracker itself, rather than using handcrafted strategies for the tracker to passively accept reference frames. This empowers the tracker with an autonomous perception of reference cues, helping it adapt to target changes and preventing tracking drift. The main contributions of this work are as follows:

- We propose a novel token context-aware tracking pipeline name LMTrack. based on a Token Context Memory module. Unlike existing tracking methods with frame-level context, LMTrack automatically collect and update high-quality token-context for visual tracking
- We introduce an effective unidirectional attention mechanism to establish dependencies between reference tokens and search frame in a unidirectional propagation manner, enabling robust cross-frame association and localization.
- Our approach achieves a new state-of-the-art tracking results on five visual tracking benchmarks, including LaSOT, TrackingNet, GOT10K, LaSOT$_{\mathrm{ext}}$, VOT2020.

Related Work

Traditional Tracking Framework. Visual object tracking has evolved significantly over the years, with traditional methods primarily relying on initial template approaches. Early methods (Bertinetto et al. 2016; Xu et al. 2020; Li et al. 2019; Chen et al. 2020) utilized Siamese networks to match the initial target template against candidate regions in subsequent frames. Although these methods effectively avoided tracker drift, they struggled to adapt to significant changes in the target's appearance. In recent years, the introduction of the transformer (Vaswani et al. 2017) enables trackers (Chen et al. 2021; Ye et al. 2022) to enhance feature representation and matching capabilities, yet they continued to rely heavily on the initial template, limiting long-term tracking in ever-changing environments, which often requires addressing difficult target appearance issue. In contrast to these methods, we reformulate the object tracking as an important token collection task and aim to extend existing tracker to efficiently exploit the target temporal context.

Temporal Context in Visual Tracking To handle the various appearance issues, many trackers have formulated the visual tracking issue as an online learning issue, in which the target appearance is adaptively updated using the temporal context of the previous frames. UpdateNet (Zhang et al. 2019) utilizes a custom network to fuse accumulated templates and generate a weighted updated template feature for visual tracking. ATOM (Danelljan et al. 2019) adds IoU prediction branches to constrain template selection. STMTrack (Fu et al. 2021) updates dynamic templates at a fixed interval to counteract changes in the target appearance. STARK (Yan et al. 2021a) and Mixformer (Cui et al. 2022) adopt an additional scoring head to verify whether the template contains the target, as the basis for selecting the template. SeqTrack (Chen et al. 2023) introduced a likelihood-based strategy that adopts the likelihood of generated tokens to select dynamic templates. RFGM (Zhou et al. 2024) selects the most appropriate template patches for the current search region, allowing for adaptation to variations.

Nevertheless, the above tracking methods still suffer from the following limitations: (1) Most methods are designed to crop and update templates based on the bounding box. However, during online learning, they often incorporate a significant amount of noise or background, as the object typically does not occupy the entire bounding box. (2) Although they explore the temporal context to some extent, they update the template using manual approaches or additional discriminator models, failing to distinguish which contexts are essential for tracking. To overcome these limitations, we propose LMTrack based on the *less-is-more* principle, which autonomously analyzes the importance distribution of all reference tokens, collecting and updating important target tokens as reference cues for subsequent video frames.

Approach

We propose a novel Token Context-Aware Tracking (LMTrack) based on the *less-is-more* principle. As depicted in Fig. 2, LMTrack comprises two key components: a novel Token Context Memory (TCM) module and an efficient unidirectional attention mechanism. This section begins with a brief introduction to our LMTrack framework, then introduction of the proposed token context memory module and the unidirectional attention mechanism.

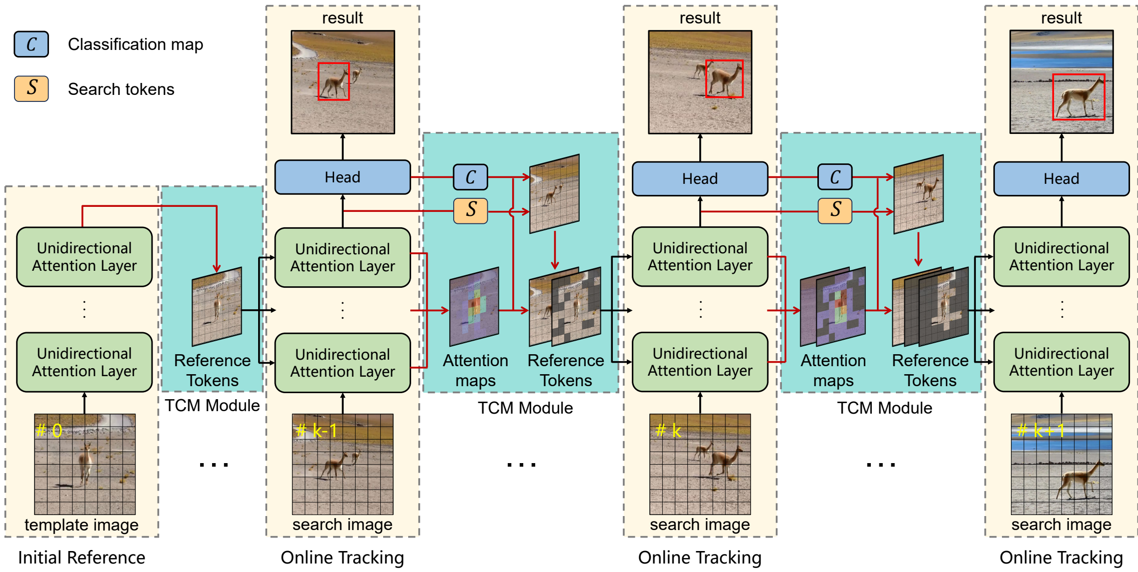

Figure 2: **The architecture of the LMTrack**. LMTrack consists of three parts, a backbone with unidirectional attention, a token context memory (TCM) module, and a prediction head. The input of tracking pipeline contains a video frame and reference tokens being collected. The TCM module utilizes classification maps and attention matrices to analyze the importance distribution of all reference tokens, then collect the important reference tokens according to this distribution.

Framework Formulation

To provide a comprehensive understanding of our LMTrack, it is essential to introduce our autoregressive token context-aware tracking framework. Unlike previous approaches that only use the template image as a reference, our framework constructs the set of reference tokens based on all video frames for visual object tracking. Therefore, we abandon the traditional approach of inputting image pairs consisting of template and search frames, opting instead for a single-frame input method. In this way, the corresponding reference tokens are adaptively collected and updated for each frame. In other words, we no longer strictly distinguish between template and search frames, instead, we treat each video frame equally, applying the same processes for target localization and reference token collection. Specifically, we initialize the reference tokens R_0 using the template I_0, which shares the same backbone g as the search frame $I_t (t > 0)$. During the tracking process f at time step t, the input consists of a search frame I_t and the reference tokens R_{t-1} being collected, while the output includes the predicted bounding box B_t and the new reference tokens R_t. The autoregressive tracking process f is formalized as follows:

$$\begin{aligned} R_0 &= f(I_0, \emptyset), t = 0, \\ B_t, R_t &= f(I_t, R_{t-1}) = f(I_t, f(I_{t-1}, R_{t-2})), t > 0. \end{aligned} \tag{1}$$

Specifically, LMTrack utilizes the token context memory module to incrementally collect the relevant reference tokens from the initial frame to the current frame, which are then used as reference context for subsequent frames. During this process, LMTrack discards irrelevant reference tokens, regardless of whether they originate from the initial or search frames. Our tracking framework consists of three components: a backbone network g with a unidirectional attention mechanism, a prediction head h, and a reference context memory module. The representation process at time step t is as follows:

$$\begin{aligned} S_t, A_t &= g(I_t, R_t), \\ B_t, C_t &= h(S_t), \\ R_t &= TCM(S_t, C_t, A_t, R_{t-1}). \end{aligned} \tag{2}$$

In this formulation, S represents the search tokens processed by the backbone, while A denotes the attention matrix that captures the distribution of importance between the reference tokens R and the search tokens S. The backbone function g employs the unidirectional attention mechanism rather than the traditional attention mechanism. The classification map C corresponds to each search token in S.

Token Context-aware Tracking Pipeline

In this section, we first introduce the Token Context Memory (TCM) module, which employs classification maps and attention maps to analyze the importance distribution of all reference tokens. Next, we present a unidirectional attention mechanism to effectively capture this importance distribution and enhance the efficiency of feature fusion. These components are designed to automatically learn high-quality reference tokens for visual tracking and consistently focus on these crucial tokens.

Token Context Memory (TCM) Module As illustrated in Fig. 3, the token context memory module is divided into three steps: (1) Collect the important tokens from the existing reference tokens based on the classification map and the attention matrix; (2) Integrate the predicted classification map into the search tokens to be used as part of the reference tokens; (3) Update the reference tokens from steps (1) and (2) for subsequent tracking.

STEP 1: Collect the important tokens from reference tokens. Unlike trackers such as (Chen et al. 2023; Cui et al. 2022), which update template images based on temporal distance, LMTrack gathers fine-grained, high-quality tokens from reference tokens that contain background and outdated target of redundant information according to their importance distribution. Fortunately, leveraging the powerful correlation calculations of the attention mechanism in the transformer architecture, LMTrack can directly sample the cross-attention maps in each encoder layer, which primarily control the impact of the references within the encoder. This is combined with the tracking results to serve as a standard for collecting important reference tokens. Formally, we formulate the collection process as follows:

$$\begin{aligned} W &= \sum_{j=1}^{L} A^j \times C, \\ R' &= Topk(Rank(R, W)), \end{aligned} \tag{3}$$

where A^j represents the cross-attention matrix between the reference tokens R and search tokens S^{j-1} in the j-th transformer layer, capturing the importance distribution of the current search tokens for reference tokens. The C denotes the classification score map, reflecting the target distribution of the search tokens S. LMTrack utilizes the C and A to assess the importance distribution of the reference tokens R. This process highlights the influence of each reference token R on the target distribution in the search token S and facilitates the differentiation of the importance of each reference token. Specifically, LMTrack uses $A \times C$ as the refined metric for importance distribution, as opposed to merely summing the attention weights of all search tokens for each reference token. Subsequently, the importance distribution is aggregated across each encoder layer. The W signifies the importance distribution of each reference token R relative to all search tokens S. LMTrack retains the reference tokens R corresponding to the k largest W as R'. In the encoder layer, multiple importance distributions W^m are generated due to multi-head attention, where $m = 1, ..., M$ and M is the number of attention heads. LMTrack averages these importance distributions across all heads to obtain an overall importance score for the reference tokens R.

STEP 2: Integrate the classification map and the search tokens. To enhance the representational power of the context memory, it is crucial to fully leverage the prediction results for the generated reference tokens. Specifically, LMTrack incorporates the category vector of the target, E_{target}, and the background, $E_{background}$. This integration is based on a binary classification score $C_{bin} \in [0, 1]$ obtained from C, S is the output of the last encoder layer in the backbone g and is used as a part of reference tokens R in subsequent tracking. This method enables the tracker to access not only potential reference tokens but also previous tracking results, thereby providing more comprehensive information than the original image alone. The integration process is formally de-

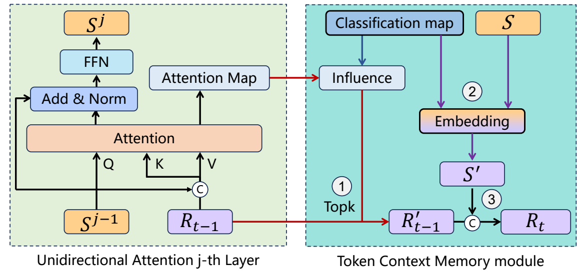

Figure 3: The unidirectional attention mechanism within the encoder layer is integrated with the token context memory module. The inputs to the unidirectional attention include search tokens and reference tokens. The token context memory module uses the attention map from unidirectional attention and predicted results to aggregate reference tokens.

fined as follows:

$$S' = S + C_{bin}E_{target} + (1 - C_{bin})E_{background}. \tag{4}$$

STEP 3: Update the reference tokens. The LMTrack obtains the reference tokens R'_{t-1} identified in step (1) and the search features S'_t that integrate the current tracking results from step (2). These components are then merged to form the reference tokens R_t and update the R_{t-1} for subsequent tracking. By continuously repeating these steps, the tracker constantly aggregates reference tokens that are valuable to the tracking process. Following this new modeling approach, LMTrack adheres to the principle of **less-is-more**, achieving an autoregressive context-aware framework by selectively retaining fewer important reference tokens.

Unidirectional Attention Mechanism The execution process of the TCM module shows that the accuracy of the importance distribution is crucial for effectively collecting the correct reference tokens. To enhance feature fusion efficiency and ensure the accuracy of the importance distribution, we employ a novel unidirectional attention mechanism to integrate reference features into search features, as illustrated in Fig. 3. Compared to traditional attention fusion methods, this approach effectively prevents changes in the context representation of reference tokens caused by the influence of search tokens. For the backbone g of a unidirectional attention encoder layer in Eq. 2, the input consists of the S (search tokens) from the previous encoder layer and the R (reference tokens) updated in the previous tracking process. The operation is shown in the following formula:

$$\begin{aligned} S &= Softmax\left(\frac{Q_s[K_r; K_s]^T}{\sqrt{d_k}}\right)[V_r; V_s] \\ &= Softmax\left(\frac{[Q_s {K_r}^T; Q_s {K_s}^T]}{\sqrt{d_k}}\right)[V_r; V_s], \\ &= [A_r; A_s][V_r; V_s]. \end{aligned} \tag{5}$$

In the j-th unidirectional attention encoder layer, the previous output S^{j-1} is projected into query, key, and value matrices Q_s, K_s, and V_s, respectively, while the reference

Method	Type	Resolution	Params	FLOPs	Speed	Device
SeqTrack	ViT-B	384×384	89M	148G	$21fps$	3090
LMTrack	ViT-B	384×384	92M	69G	$47fps$	3090

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

tokens R are projected into key and value matrices K_r and V_r. Each unidirectional attention encoder layer uses the same R within a single time step. The matrix A_r represents the cross-attention matrix employed in the TCM module in Eq. 3. This unidirectional attention mechanism ensures that only the reference tokens R affect the search tokens S, thereby maintaining a consistent representation of the reference tokens R and reducing unnecessary computations.

It is worth noting that we have implemented a context-aware token mechanism to automatically gather high-quality reference tokens using the TCM module. This process enhances the collection of crucial reference tokens for LMTrack. During both training and inference, the TCM and the unidirectional attention mechanism work synergistically. The TCM improves reference token extraction within the unidirectional attention encoder by collecting appropriate tokens, while the unidirectional attention mechanism generates precise attention maps that benefit the effective collection of reference tokens.

Head and Loss

The output features S from the encoder are input into a Fully Convolutional Network (FCN), which consists of L stacked Conv-BN-ReLU layers for each output. The output of the FCN includes target classification score map $\mathbb{R}^{\frac{H_x}{P} \times \frac{W_x}{P}}$, offset size $\mathbb{R}^{2\times \frac{H_x}{P} \times \frac{W_x}{P}}$ for compensating for discretization errors caused by reduced resolution, and normalized bounding box dimensions $\mathbb{R}^{2\times \frac{H_x}{P} \times \frac{W_x}{P}}$.

During training, both classification loss and regression loss are simultaneously employed. We utilize weighted focal loss (Lin et al. 2017) for classification. For bounding box regression, we use predicted bounding boxes, L_1 loss, and generalized IoU loss (Rezatofighi et al. 2019). The total loss function is defined as:

$$L = L_{cls} + \lambda_{iou} L_{iou} + \lambda_{L1} L_1, \tag{6}$$

where $\lambda_{iou} = 2$ and $\lambda_{L_1} = 5$.

Experiments

Implementation Details

Training. We use ViT-base (Dosovitskiy et al. 2021) model as the visual encoder. The training data includes LaSOT (Fan et al. 2019), GOT-10k (Huang, Zhao, and Huang 2021), TrackingNet (Müller et al. 2018), and COCO (Lin et al. 2014). We employ the AdamW to optimize the network parameters with initial learning rate of 4×10^{-5} for the backbone, 4×10^{-4} for the rest, and set the weight decay to 10^{-4}. We set the training epochs to 300 epochs. $60,000$ search images are randomly sampled in each epoch. The learning rate drops by a factor of 10 after 240 epochs. The model is conducted on a server with two 80GB Tesla A100 GPUs, using a batch size of 16, where each batch consists of four search images and one template image.

Inference. In the initial stage of tracking, we use the first template to initiate the reference tokens. LMTrack records the importance distribution for all reference tokens in each frame. It defaults to a reference update check every 400 frames and resets the importance distribution accordingly. When the sampled tokens from the search exceed the upper limit of the reference token length, we collect important reference tokens according to the importance distribution. The default maximum length of the reference tokens is twice the length of the search tokens. After collection, the length of the reference tokens is maintained at the initial length.

Comparison with State-of-the-Art Trackers

We demonstrate the effectiveness of LMTrack, we compare them with state-of-the-art (SOTA) trackers on seven different benchmarks, including GOT-10K (Huang, Zhao, and Huang 2021), TrackingNet (Müller et al. 2018), LaSOT (Fan et al. 2019), $\mathrm{LaSOText}$ (Fan et al. 2021), VOT2020 (Kristan, Leonardis, and et.al 2020).

GOT-10K. GOT-10K (Huang, Zhao, and Huang 2021) dataset is an extensive dataset comprising over 10,000 video segments, with 180 segments designated for testing. Following the official requirements, we only use the GOT-10k training set to train our model and evaluated the test results. As reported in Tab. 2, LMTrack has achieved a remarkable state-of-the-art performance 80.1% AO when compared to the previous best performance ARTrackV2 77.5% AO. These results demonstrate that one benefit of our LMTrack comes from token context-aware trackinng pipeline, which effect collects the token context during tracking.

TrackingNet. TrackingNet (Müller et al. 2018) is a large-scale dataset containing 511 videos and boasts a collection of over 30,000 videos with more than 14 million densely annotated bounding boxes. We evaluated $\mathrm{LMTrack}_3 84$ on its test set and achieved an impressive 85.7% AUC on this large-scale benchmark.

LaSOT. LaSOT (Fan et al. 2019) dataset consists of 280 videos in its test set with an average length of 2448 frames. To assess the long-term tracking capabilities of LMTrack. $\mathrm{LMTrack}_{384}$ surpasses the most of tracker, achieving a 73.2% AUC. These results demonstrate that the TCM module can capture long-time contextual cues more efficiently.

$\mathrm{LaSOT}_{\mathrm{ext}}$. $\mathrm{LaSOT}_{\mathrm{ext}}$ (Fan et al. 2021) is an extended subset of LaSOT that includes 150 additional videos from 15 new categories. These new sequences introduce challenging tracking scenarios, such as occlusions and fast-moving small objects. LMTrack gets a 53.6% AUC, 64.7% P_{Norm} and 61.5% P, outperforming the ARTrackV2 by 0.7%, 1.3%, 2.4%, respectively. This demonstrates the robustness of LMTrack in handling these difficult scenarios.

VOT2020. VOT2020 (Kristan, Leonardis, and et.al 2020) contains 60 challenging sequences and uses binary segmentation masks as the groundtruth. We use Alpha-Refine (Yan et al. 2021b) as a post-processing network to predict segmentation masks. As shown in Tab. 3, $\mathrm{LMTrack}_{256}$ and

Method	GOT-10K*			LaSOT			$\text{LaSOT}_{\text{ext}}$			TrackingNet		
	AO	$\text{SR}_{0.5}$	$\text{SR}_{0.75}$	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P
SiamPRN++ (Li et al. 2019)	51.7	61.6	32.5	49.6	56.9	49.1	34.0	41.6	39.6	73.3	80.0	69.4
DiMP (Bhat et al. 2019)	61.1	71.7	49.2	56.9	65.0	56.7	39.2	47.6	45.1	74.0	80.1	68.7
SiamRCNN (Voigtlaender et al. 2020)	64.9	72.8	59.7	64.8	72.2	-	-	-	-	81.2	85.4	80.0
Ocean (Zhang et al. 2020)	61.1	72.1	47.3	56.0	65.1	56.6	-	-	-	-	-	-
STMTrack (Fu et al. 2021)	64.2	73.7	57.5	60.6	69.3	63.3	-	-	-	80.3	85.1	76.7
TrDiMP (Wang et al. 2021)	67.1	77.7	58.3	63.9	-	61.4	-	-	-	78.4	83.3	73.1
TransT (Chen et al. 2021)	67.1	76.8	60.9	64.9	73.8	69.0	-	-	-	81.4	86.7	80.3
Stark (Yan et al. 2021a)	68.8	78.1	64.1	67.1	77.0	-	-	-	-	82.0	86.9	-
KeepTrack (Mayer et al. 2021)	-	-	-	67.1	77.2	70.2	48.2	-	-	-	-	-
SBT-B (Xie et al. 2022)	69.9	80.4	63.6	65.9	-	70.0	-	-	-	-	-	-
Mixformer (Cui et al. 2022)	70.7	80.0	67.8	69.2	78.7	74.7	-	-	-	83.1	88.1	81.6
TransInMo (Guo et al. 2022)	-	-	-	65.7	76.0	70.7	-	-	-	81.7	-	-
OSTrack_{384} (Ye et al. 2022)	73.7	83.2	70.8	71.1	81.1	77.6	50.5	61.3	57.6	83.9	88.5	83.2
AiATrack (Gao et al. 2022)	69.6	80.0	63.2	69.0	79.4	73.8	47.7	55.6	55.4	82.7	87.8	80.4
SeqTrack_{384} (Chen et al. 2023)	74.5	84.3	71.4	71.5	81.1	77.8	50.5	61.6	57.5	83.9	88.8	83.6
GRM (Gao, Zhou, and Zhang 2023)	73.4	82.9	70.4	69.9	79.3	75.8	-	-	-	84.0	88.7	83.3
VideoTrack (Xie et al. 2023)	72.9	81.9	69.8	70.2	-	76.4	-	-	-	83.8	88.7	83.1
ARTrack_{384} (Xing et al. 2023)	75.5	84.3	74.3	72.6	81.7	79.1	51.9	62.0	58.5	85.1	89.1	84.8
ODTrack_{384} (Zheng et al. 2024)	77.0	87.9	75.1	**73.2**	83.2	80.6	52.4	63.9	60.1	85.1	**90.1**	84.9
HIPTrack_{384} (Cai, Liu, and Wang 2024)	77.4	88.0	74.5	72.7	82.9	79.5	53.0	64.3	60.6	84.5	89.1	83.8
AQATrack (Xie et al. 2024)	76.0	85.2	74.9	72.7	82.9	80.2	52.7	64.2	60.8	84.8	89.3	84.3
ARTrackV2 (Bai et al. 2024)	77.5	86.0	75.5	73.0	82.0	79.6	52.9	63.4	59.1	**85.7**	89.8	**85.5**
LMTrack_{256}	76.3	87.1	73.9	69.8	79.2	76.3	49.0	59.6	55.8	84.2	89.0	82.8
LMTrack_{384}	**80.1**	**91.5**	**79.0**	**73.2**	**83.4**	**81.0**	**53.6**	**64.7**	**61.5**	**85.7**	89.9	84.7

Table 2: Comparison with state-of-the-arts on four popular benchmarks: GOT-10K (Huang, Zhao, and Huang 2021), LaSOT (Fan et al. 2019), $\text{LaSOT}_{\text{ext}}$ (Fan et al. 2021), and TrackingNet (Müller et al. 2018). * denotes for trackers only trained on GOT-10K. The best two results are in **bold** and underline, respectively.

	STM	SiamMask	Ocean	D3S	AlphaRef	Ocean+	STARK	SBT	Mixformer	SeqTrack	ODTrack_{384}	**LMTrack_{256}**	**LMTrack_{384}**
EAO(↑)	0.308	0.321	0.430	0.439	0.482	0.491	0.505	0.515	0.535	0.522	0.581	0.550	**0.586**
Accuracy(↑)	0.751	0.624	0.693	0.699	0.754	0.685	0.759	0.752	0.761	-	**0.764**	0.752	0.753
Robustness(↑)	0.574	0.648	0.754	0.769	0.777	0.842	0.819	0.825	0.854	-	0.877	0.852	**0.895**

Table 3: State-of-the-art comparison on VOT2020 (Kristan, Leonardis, and et.al 2020). The best two results are in **bold** and underline, respectively.

LMTrack_{384} achieve the EAO results of 55% and 58.6% on mask evaluations, respectively, demonstrating the effectiveness of the token context-aware approach.

#	Attention	autoregressive	Update	AO(%)
1	bidirectional	×	-	73.0
2	unidirectional	×	-	73.9
3	unidirectional	×	update template	74.1
4	unidirectional	×	TCM	75.0
5	unidirectional	✓	update template	75.6
6	unidirectional	✓	TCM	76.3

Table 4: Ablation experiment about LMTrack in GOT-10K.

Ablation and Analysis

In this section, we perform a detailed analysis of the key components of LMTrack_{256}. In all our experimental studies, we adhere to the GOT-10K test protocol.

The Unidirectional Attention. To evaluate the impact of the unidirectional attention mechanism described in Eq. 5, we conducted experiments comparing different attention mechanisms, as shown in Tab. 4. The bidirectional attention method processes both the search and template images simultaneously, whereas the unidirectional attention method only takes the search image and reference tokens from the initial template image as inputs. Observations from the first and second rows indicate that the unidirectional attention mechanism prevents noise from propagating from the search to the reference, resulting in a 0.9% increase in average overlap (AO). Additionally, unidirectional attention improves fusion efficiency. As seen in Tab. 1, unidirectional attention significantly enhances inference speed when using the same template/reference token sizes. This shows that unidirectional attention not only prevents noise propagation but also eliminates duplicate modeling of template features.

Autoregressive Tracking. We compare the different tracking method on the performance. Traditional methods require cropping the template based on previous results and extracting features with the backbone at each update. This approach uses only $R_0 = f(I_0, \emptyset), t = 0$ and does not employ $B_t, R_t = f(I_t, R_{t-1}), t > 0$. As shown in Tab. 4 (rows three to six), autoregressive feature extraction outperforms

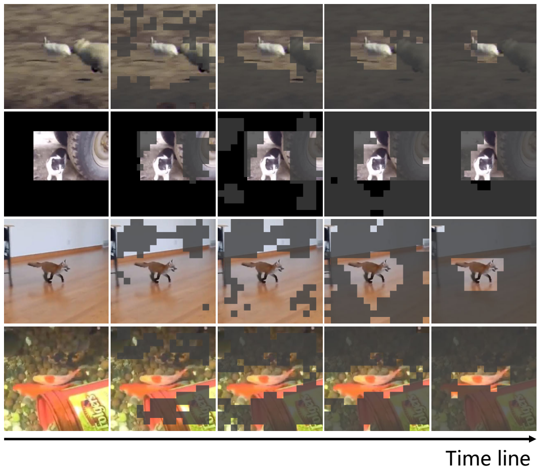

Figure 4: The TCM module visualization shows the collection of significant reference tokens over time. The highlighted part indicates retained tokens for ongoing tracking, illustrating the module's ability to filter out invalid tokens.

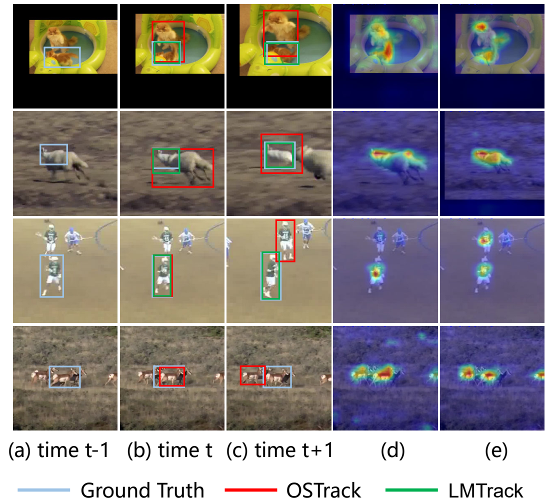

Figure 5: Comparison of the response between LMTrackand OSTrack. (a)-(c): Search regions with predict boxes. (d): Attention maps of LMTrack. (e): Attention maps of OSTrack.

traditional methods, achieving a 1.5% and 1.3% improvement in AO over the template update method and the TCM module, respectively. Traditional methods lack reference tokens for accurately modeling target information in subsequent templates, limiting their ability to effectively disseminate target information throughout the video. In contrast, the autoregressive method leverages reference tokens updated in previous tracking steps, adapting more effectively to subsequent tracking processes and eliminating the need for redundant operations on new template images.

Token Context Memory Module. The Token Context Memory (TCM) module is designed to enhance reference token representation by adhering to a *less-is-more* principle, selectively retaining a less number of important reference tokens. This selective retention strategy allows for more precise updates of the reference tokens and updates each reference token independently, instead of modifying the entire template image. To evaluate the impact of various update strategies, we adopt LMTrack as our default configuration. The results are shown in Tab. 4 (rows three to six), demonstrating that employing more fine-grained operations achieves a 0.7% and 0.9% improvement in AO with and without autoregressive tracking, respectively. This shows that LMTrack is able to constantly aggregate reference tokens that are valuable for the tracking process.

Visualization and Qualitative Analysis

Visualization of the TCM Module: Fig. 5 depicts the process by which our TCM module extracts significant reference tokens from the same frame over time. As time progresses, most background tokens become less significant, with fewer reference tokens primarily describing the target's appearance. Our autoregressive tracking method effectively captures the reference tokens associated with the target, even in challenging circumstances that involve variations in appearance and potential distractions.This illustrates a notable adaptability in accurately identifying the reference tokens of the tracking target.

Compare Response Between reference tokens and Template: When faced with these difficult scenarios, the template image may be unreliable. To address these problems, our method uses reference tokens instead of the original template image to guide object tracking. In Fig. 5, we compare the attention maps generated by the reference tokens of LMTrack and the template of OSTrack (Ye et al. 2022). Unlike OSTrack, which uses the template image for guidance, LMTrack employs historical search features to generate reference tokens and collects important reference tokens via the TCM module, effectively resisting various appearances and distractions. By observing tracking results frame-by-frame and attention response maps, we find that the attention of OSTrack becomes distracted when similar objects are present in the search image, leading to erroneous tracking. In contrast, LMTrack can maintain focus on the target by using reference tokens that have proven important in previous tracking steps to keep attention on the target.

Conclusion

This study introduces the Token Context-Aware Tracker (LMTrack), which is based on the principle that less tokens with tracker attention play a more important role in the results. LMTrack comprises two key components: The Token Context Memory (TCM) module and the unidirectional attention mechanism in the encoder layer. The TCM module analyzes the importance distribution of all reference tokens, collecting, attending to, and updating the important ones. Additionally, LMTrack adopts the unidirectional attention mechanism to establish dependencies between reference tokens and search frame in a unidirectional propagation manner. Thus, our approach discards the traditional frame-level context and achieves a fine-grained token-level context to represent important reference cues across time steps.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.U23A20383 and 62472109), 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), and the Innovation Project of Guangxi Graduate Education (YCSW2024147).

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*, 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 *ECCV Workshops*, 850–865.

Bhat, G.; Danelljan, M.; Gool, L. V.; and Timofte, R. 2019. Learning Discriminative Model Prediction for Tracking. In *ICCV*, 6181–6190.

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*, 19258–19267.

Chen, X.; Peng, H.; Wang, D.; Lu, H.; and Hu, H. 2023. SeqTrack: Sequence to Sequence Learning for Visual Object Tracking. *CVPR*, abs/2304.14394.

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.

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

Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2019. ATOM: Accurate Tracking by Overlap Maximization. In *CVPR*, 4660–4669.

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.; 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. *CVPR*, abs/2303.16580.

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

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.

Kristan, M.; Leonardis, A.; and et.al. 2020. The Eighth Visual Object Tracking VOT2020 Challenge Results. In *ECCV Workshops (5)*, volume 12539 of *Lecture Notes in Computer Science*, 547–601. Springer.

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, 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.

Mayer, C.; Danelljan, M.; Paudel, D. P.; and Gool, L. V. 2021. Learning Target Candidate Association to Keep Track of What Not to Track. In *ICCV*, 13424–13434. IEEE.

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.

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.

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

Voigtlaender, P.; Luiten, J.; Torr, P. H. S.; and Leibe, B. 2020. Siam R-CNN: Visual Tracking by Re-Detection. In *CVPR*, 6577–6587.

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

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, F.; Wang, C.; Wang, G.; Cao, Y.; Yang, W.; and Zeng, W. 2022. Correlation-Aware Deep Tracking. In *CVPR*, 8741–8750.

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, W.; Yifan, B.; Yongchao, Z.; Dahu, S.; and Yihong, G. 2023. Autoregressive Visual Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9697–9706.

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.

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

Yan, B.; Zhang, X.; Wang, D.; Lu, H.; and Yang, X. 2021b. Alpha-Refine: Boosting Tracking Performance by Precise Bounding Box Estimation. In *CVPR*, 5289–5298. Computer Vision Foundation / IEEE.

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.

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

Zhang, Z.; 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.

Zhou, X.; Guo, P.; Hong, L.; Li, J.; Zhang, W.; Ge, W.; and Zhang, W. 2024. Reading relevant feature from global representation memory for visual object tracking. *Advances in Neural Information Processing Systems*, 36.