High-Performance Discriminative Tracking with Spatio-Temporal Template Fusion

Xuedong He*
Huiying Xu*
hexuedong@zjnu.edu.cn
xhy@zjnu.edu.cn
School of Computer Science and Technology, Zhejiang Normal University
Jinhua, China

Xinzhong Zhu
zxz@zjnu.edu.cnn
School of Computer Science and Technology, Zhejiang Normal University
Jinhua, China
Research Institute of Hangzhou Artificial Intelligence, Zhejiang Normal University
Hangzhou, China
Beijing Geekplus Technology Co., Ltd
Beijing, China

Hongbo Li
jason.li@geekplus.com
Beijing Geekplus Technology Co., Ltd
Beijing, China

Abstract

The current one-stream tracking framework has received far-reaching attention for its significant improvement in tracking performance, yet it is essentially an extension of Siamese trackers. However, the one-stream framework of discriminative trackers has not been effectively exploited, still using separate feature extraction and model prediction. Therefore, this article aims to implement a one-stream learning strategy for feature extraction and model prediction under the discriminative tracking framework. To this end, we have leveraged the prevailing Vision Transformer and Vision Mamba backbones to achieve our motivation. Moreover, we innovatively combine templates with discriminative tracking methods to enhance the ability of target-aware feature learning, and further propose the attention fusion module to implement spatiotemporal template fusion, which can enhance the adaptability of the tracking model to dynamic changes of targets. The experiments on multiple popular tracking benchmarks have demonstrated that our proposed tracking architecture has superior tracking performance. Concisely, our tracker obtains an AUC of 73.3% on LaSOT dataset, and an AO of 78.2% on GOT-10k dataset. The code, raw results, and trained models are available at https://github.com/hexdjx/VisTrack.

CCS Concepts

• **Computing methodologies** → **Tracking**.

Keywords

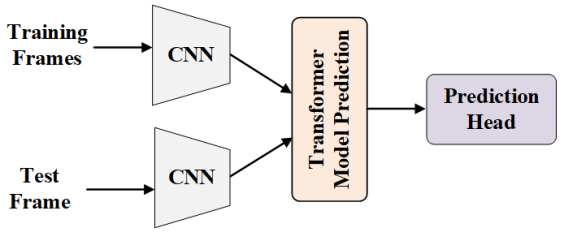
Visual Tracking, Discriminative Model Prediction, Vision Transformer, Vision Mamba, Template Fusion

*Huiying Xu and Xuedong He are the corresponding authors.

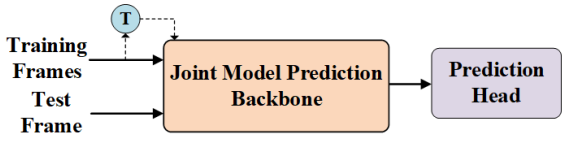
MM '25, October 27–31, 2025, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755721

ACM Reference Format:
Xuedong He, Huiying Xu, Xinzhong Zhu, and Hongbo Li. 2025. High-Performance Discriminative Tracking with Spatio-Temporal Template Fusion. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3755721



(a) Pipeline of the Discriminative Transformer Tracking



(b) Pipeline of our proposed Joint Discriminative Tracking with Template Embedding

Figure 1: Simplified illustration of our proposed framework, which is a novel and concise one-stream discriminative tracking compared to the two-stream discriminative methods.

1 Introduction

Visual object tracking is the process of tracking a specific object in a video, with only the first frame of the target state provided, the characteristic of which is that the tracked object is single target and its category is agnostic. In the past decade, discriminative filters and Siamese networks [22] have led the development of visual object tracking. Due to the complexity of tracking scenarios, object tracking remains active at the forefront of research.

With the development of deep learning, mainly Convolutional Neural Network (CNN) [20] and Vision Transformer (ViT) [11], the tracking performances of discriminative trackers [2, 30, 36, 45] and Siamese trackers [7, 24, 43] have been greatly improved. Siamese trackers can be divided into feature extraction, feature fusion, and prediction head. In the beginning, CNN is used as the feature extractor, and correlation operations [24] are used to fuse template and search region features to find the most similar targets. However, template features are generally unchanged during the tracking process. Nowadays, some works[6, 9, 44] adopt ViT backbone to perform joint feature extraction and feature fusion learning, which is called a one-stream tracking pipeline. Especially, anchor-based or anchor-free prediction heads [9, 24] also provide support for the improvement of tracking performance.

The Siamese trackers mentioned above use template matching to find the most similar target object from the search region, and its template contains little target information while discriminative trackers [2, 30] are intended to train a discriminative model prediction to recognize the real target from background information, especially excellent DiMP [2] and ToMP [30] acquire the Discriminative Correlation Filter (DCF) target model in end-to-end training. DCF-based trackers have been developed for many years and have achieved promising results [22], and some discriminative trackers (*e.g.,* TrDiMP [36], DTT [45], and ToMP [30]) have already adopted the Transformer structure for learning the discriminative target model. However, their current tracking performance is largely surpassed by Siamese trackers using one stream Transformer framework. Therefore, we have identified two important reasons from the analysis of ToMP and existing one-stream Siamese trackers. (1) Feature representation: the existing ToMP still uses ResNet as the backbone for feature extraction, and its expression ability is not as good as that of the ViT backbone. Moreover, using pre-trained network features is not as good as training on object tracking datasets (*e.g.,* LaSOT [13], TrackingNet [32], and GOT-10k [21]). These two aspects greatly improve the performance of Siamese trackers on these tracking datasets. (2) Online model update: online model update has always been a standard component of DCF trackers, which effectively utilizes changing target information to maintain robust tracking. Nowadays, Siamese-Transformer trackers use a score prediction head [9, 14, 43] to attach the dynamic template, and implement feature interaction under the Transformer architecture, greatly improving the disadvantage of fixed templates in early Siamese trackers. Some works [42, 46] use video clips to construct a token propagation mechanism. Therefore, it is necessary to properly judge the prediction results to update the target template.

Transformer-based trackers have been developed in recent years and have achieved promising results, their current tracking performance is largely attributed to the learning ability of the Transformer. Especially, one-stream Siamese tracking framework fully realizes efficient feature extraction and sufficient feature fusion. Moreover, replacing ViT backbone with Vision Mamba[47] has also been used in the Siamese tracking framework [25, 37]. According to the current survey, one-stream [6, 9, 44] and two-stream [7, 43] tracking frameworks using the Siamese paradigm have been developed, while there is no one-stream method proposed for the discriminative tracking framework. As is shown in Figure 1, this paper innovatively proposes a one-stream discriminative tracking framework. Our approaches demonstrate competing performances on eight challenging benchmarks [12, 13, 21, 23, 31, 32, 38, 41].

In summary, our contributions are as follows: (1) We propose a novel and concise one-stream discriminative tracking architecture, which uses our devised target model prediction module and joint learning backbone to realize the integration of feature extraction and discriminative model prediction. (2) We introduce the Vision Transformer block and Vision Mamba block to construct a one-stream discriminative joint learning backbone. (3) We construct a target model prediction based the given bounding box to encode the discriminative features and propose a target template to enhance the acquisition of target-aware features. (4) Comprehensive ablative studies and comparative experiments are implemented on eight fashionable tracking benchmarks to verify the feasibility of our tracking framework.

2 Related Work

At present, most visual object tracking algorithms are implemented using deep learning architectures. Based on the taxonomy of this review [22], we mainly provide an overview of Siamese and discriminative tracking.

2.1 Siamese Tracking

Since its inception, Siamese trackers have been using learnable methods to learn target features. TransT [7] constructed a Transformer fusion module to integrate template and search branch features, and STARK [43] introduced a dynamic template to further enhance the effect of feature fusion. AiATrack [15] and CSWinTT [35] improved Transformer attention mechanism to enhance feature correlation ability. SimTrack [6] and OSTrack [44] formulated an innovative one-stream framework, which does not require a CNN feature backbone, but only uses ViT [11] to construct joint feature extraction and fusion to achieve more efficient interaction between the template and the search area. MixFormer [9] used ViT and CVT [40] (which is a variant of ViT) to jointly learn feature extraction and fusion. Similarly, it also proposed a score prediction module to update dynamic templates. GRM [16] proposed a token division module and attention masking strategy to improve one-stream Transformer tracking. ROMTrack [5] proposed a novel modeling method to model the inherent template and the hybrid template features simultaneously. HIPTrack [4] utilized historical locations and visual features to generate historical cues to enhance tracking performance. Moreover, a learnable query token [42, 46] and Vision Mamba backbone [25, 37] is adopted for constructing target context-aware learning.

2.2 Discriminative Tracking

Before deep learning, DCF has always been a hot topic for visual tracking. The proposal of DCF tracker [3] was several years earlier than that of Siamese tracker [1], and DCF trackers [10, 30] have undergone the evolution of manual features, CNN, and Transformer features. DCFs are a discriminative tracking method, which aim to distinguish the foreground target from the background region, but unlike Siamese tracking, they have a completely trainable tracking architecture. Therefore, DiMP [2] was the first learnable deep architecture to adopt the DCF paradigm, which was comparable to or

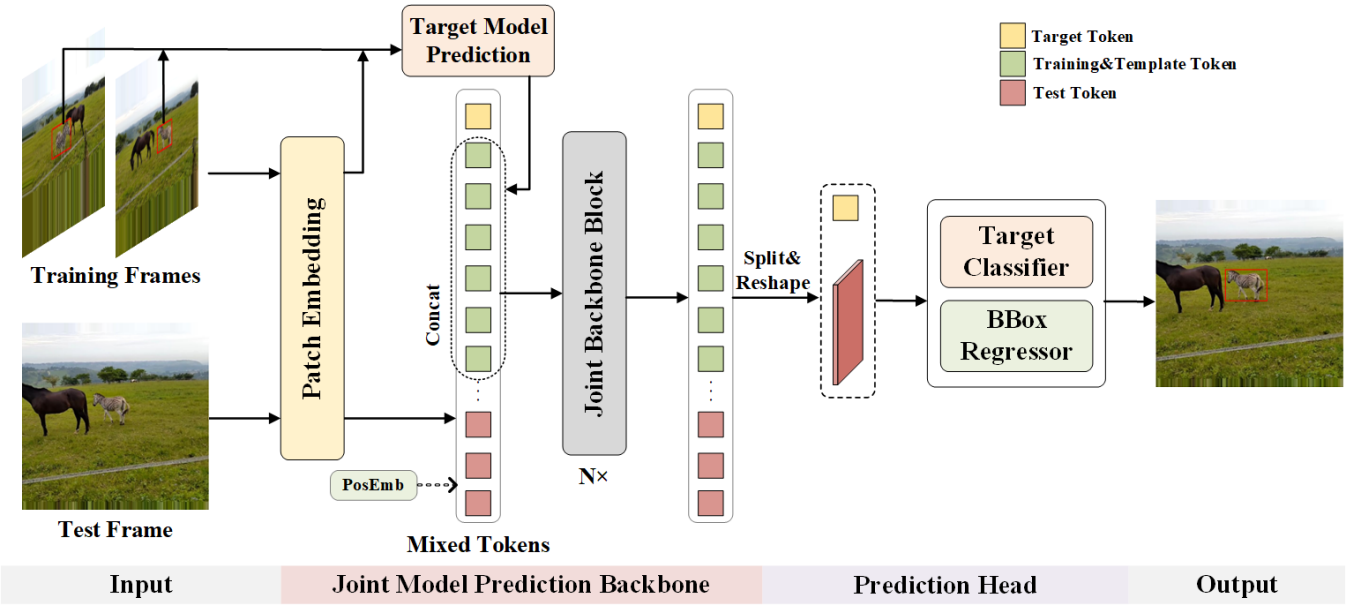


Figure 2: Overview of our one-stream discriminative Transformer tracking. Our approach constructs a joint backbone block with target model prediction to learn mixed tokens, and uses the discriminative prediction head to output the new target state.

even better than the Siamese tracker. TrDiMP [36] is the first Transformer tracker improved based on DiMP. DTT [45] proposed an encoder-decoder Transformer architecture to replace the optimized-based model predictor in DiMP. Recently, ToMP [30] proposed a Transformer model predictor to replace the optimized-based model predictor of DiMP, which can utilize Transformer to achieve interaction between training frames and test frames. In particular, DiMP uses data augmentation and a larger sample memory to store reliable samples to optimize the DCF target model, while ToMP employs a fixed initial training frame and dynamic training frame to achieve online tracking.

Summay: Whether it is discriminative or Siamese trackers, they can no longer do without the help of Transformers. The Siamese trackers adopt the template matching method and use the input description of the template and the search region. The template has a small amount of background information and is twice as small as the search region. The discriminative tracking adopts the concept of DCF to construct a discriminative target model that can output Gaussian response maps. This method uses input description of training frames and test frame, with the same resolution and background region in the training frames, aiming to learn discriminative target models. However, due to the lack of a one-stream tracking architecture, there is a significant performance gap between current discriminative tracking and one-stream Siamese tracking. This is the primary motivation behind the content of this paper.

3 Our Approach

3.1 Overview

The method in this paper aims to propose a one-stream discriminative tracking framework, which is mainly based on the two-stream discriminative ToMP tracker. As is displayed in Figure 2, we propose a joint model prediction backbone to realize joint learning of feature extraction and discriminative model prediction. Unlike the model prediction scheme implemented by ToMP using feature extraction first and then Transformer model prediction, our method is concise and completely uses the joint learning backbone, the function of which is to convert RGB images from training and test frames into deep features. Additionally, we refer to the practice of target state embedding from ToMP and propose a target template to construct the target model prediction for supervising the discriminative learning. Subsequently, we obtain the response map and *ltrb* expression using a prediction head that consists of a target classifier and a bounding box regressor, respectively. Finally, we select and transform the *ltrb* expression into the target state of the test frame by using the peak coordinates of the response map. To ensure the correct updating of the dynamic training frame (the image on the right side of the training frames in Figure 2), we only use the peak value of the response map to determine whether to update the dynamic training frame online.

3.2 Joint Feature Learning Backbone

This paper integrates the one-stream idea into the discriminative tracking framework and proposes a pure ViT and Vim backbone with a target model prediction to encode mixed tokens. Our trackers use the configuration of one-stream backbone and prediction head. Unlike one-stream Siamese tracking framework, our trackers belong to a one-stream discriminative tracker. As is displayed in Figure 2, the input branches include training frames $x_{tr} \in \mathbb{R}^{T\times H\times W\times 3}$ and test frame $x_{te} \in \mathbb{R}^{H\times W\times 3}$, T represents the number of training frames, and we can see that the image resolution of the training and testing frames is the same. Then, the training and test image frames are passed through the patch embedding PE to extract training and test tokens. The patch embedding PE is aimed at converting RGB images into token features, that is $\left[e_p^{tr}, e_p^{te}\right] = PE\left([x_{tr}, x_{te}]\right)$.

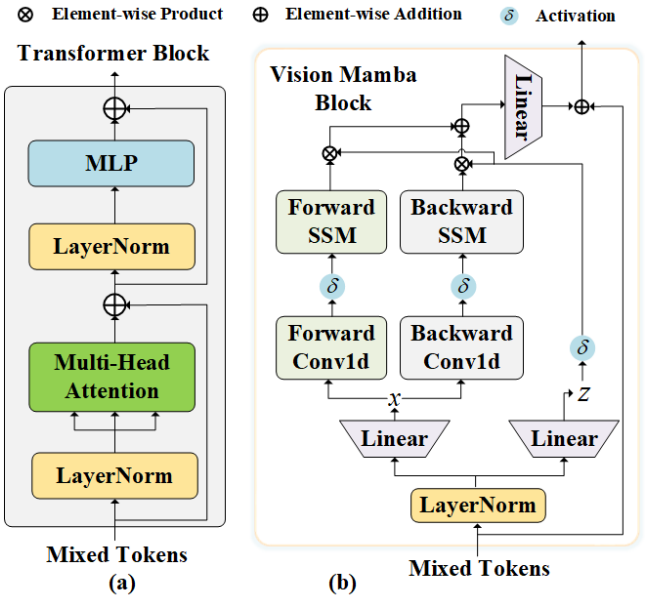


Figure 3: Illustration of Vision Transformer (ViT) block and Vision Mamba (Vim) block. We have implemented the construction of a joint feature learning backbone using ViT and Vim blocks, respectively.

Specifically, we adopt the target model prediction $\mathcal{M}$ to obtain Gaussian classification token e_c, *ltrb* regression token e_l, and target template token e_t based on bounding boxes from training frames, these e_c and e_l tokens are utilized to supervise the training of target classifier and bounding box regressor. While the e_t is used for generating enhanced target-aware features. To condense discriminative foreground target features from the training and test frames, we additionally attach a learnable foreground target token e_f to constitute the mixed tokens $e_{mix} = \left[e_f, e_t, e_p^{tr} \oplus e_c \oplus e_l, e_p^{te}\right]$.

Moreover, this foreground target token e_f is also used for encoding Gaussian classification labels. Immediately after, we pass through $N\times$ joint backbone blocks to achieve joint feature learning between the foreground target, target template, training and test tokens. We use two prevailing architectures, ViT [11] and Vim [47], to implement the joint backbone block. Here, ViT adopts the Transformer architecture with quadratic complexity, while Vim adopts the Mamba architecture with linear complexity.

Vision Transformer Block. The structure of Transformer block is provided in Figure 3a, which uses the same structure as ViT [11]. The Transformer block mainly includes multi-head attention (MHAtt), layer normalization (LN), multi-layer perceptron (MLP), and residual connection. The $N\times$ Transformer blocks can be formulated as follows:

$$\begin{aligned} z_0 &= e_{mix} \oplus E_{pos},\ E_{pos} \in \mathbb{R}^{(L+1)\times D}, \\ z_n^{att} &= z_{n-1} + MHAtt\left(LN\left(z_{n-1}\right)\right),\ \mathrm{n} = 1 \ldots N, \\ z_n^{mlp} &= z_n^{att} + MLP\left(LN\left(z_n^{att}\right)\right), \\ e_{mix}^{enc} &= LN\left(z_N^{mlp}\right). \end{aligned} \tag{1}$$

Here, E_{pos} is Sinusoidal Position Embedding used in Transformer.

Vision Mamba Block. Vim [47] proposed a novel vision backbone that utilizes bidirectional Mamba [17] blocks to learn visual representation. As shown in Figure 3b, the summation of mixed tokens e_{mix} and position encoding E_{pos} is first normalized and then passed through two diverse linear layers to obtain x and z. The core of Vim adopts the Conv1d and the state space model (SSM) module implemented by Mamba to construct a bidirectional Mamba block, which also uses layer normalization (LN), the activation function δ denotes SiLU, and element-wise product and addition, as well as residual connection to implement feature encoding. The $N\times$ Vision Mamba blocks can be formulated as follows:

$$\begin{aligned} &z_0 = e_{mix} \oplus E_{pos},\ E_{pos} \in \mathbb{R}^{(L+1)\times D}, \\ &x = Linear^x\left(LN\left(z_{n-1}\right)\right), n = 1 \ldots N, \\ &z = Linear^z\left(LN\left(z_{n-1}\right)\right), \\ &x_o = SSM_o\left(\delta\left(Conv1d_o\left(x\right)\right)\right), o \in [forward, backward], \\ &z_n = Linear\left(\left(x_{forward} \otimes \delta\left(z\right)\right) \oplus \left(x_{backward} \otimes \delta\left(z\right)\right)\right) \oplus z_{n-1}, \\ &e_{mix}^{enc} = LN\left(z_N\right). \end{aligned} \tag{2}$$

Ultimately, we split the mixed tokens e_{mix}^{enc} encoded by joint backbone blocks into the foreground target token e_f^{enc} and test token e_{te}^{enc}, which is reshaped the required dimension and inputted into the target classifier and bounding box regressor to predict response map and *ltrb* regression map, these maps are utilized to yield the new target state.

3.3 Target Model Prediction

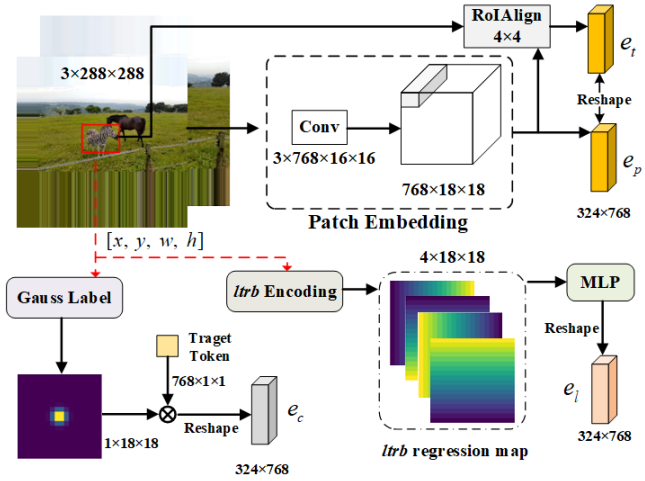


Figure 4: Structure of Target Model Prediction. The patch embedding is a convolution module with a kernel and stride of 16. For training frames, the Gauss label and *ltrb* encoding modules are utilized to acquire the supervised tokens based on bounding boxes, which are attached to patch embedding features as training tokens. Moreover, we adopt RoIAlign [33] to crop the target region as the target template token.

The previous section has introduced the one-stream joint feature learning backbone. This section mainly displays the target model prediction and analyzes how training and template tokens are generated. As is depicted in Figure 4, the training token is composed of

patch embedding, Gauss label, and *ltrb* regression features, while the template token is directly obtained from the patch embedding features. The output of these modules has undergone a 16-fold down sampling operation. Take initial image frame $x_{im} \in \mathbb{R}^{3\times288\times288}$ as an example, we first use patch embedding module to obtain image patch token, as follows:

$$e_p = \gamma\left(PE\left(x_{im}\right)\right), e_p \in \mathbb{R}^{324\times768}. \tag{3}$$

Here, PE denotes the patch embedding module, γ denotes the reshape operation. Gauss label and *ltrb* encoding demand the given target bounding box $[x, y, w, h]$, the bounding box is converted to obtain *ltrb* regression map l_r and Gaussian classification label y_c. The *ltrb* distance map l_r is mapped by an MLP layer ψ to get a *ltrb* token $e_l = \gamma\left(\psi\left(l_r\right)\right)$. The Gaussian classification label y_c is element-wise multiplied by the foreground target token e_f to yield a Gauss label token $e_c = \gamma\left(y_c \otimes e_f\right)$. Finally, we integrate *ltrb* and Gaussian label token into the feature token through element-wise addition to supervise the learning of the training frame feature. Attentively, the test frame features only include the feature token, as follows:

$$\begin{aligned} e_{tr} &= e_p^{tr} \oplus e_l \oplus e_c, \\ e_{te} &= e_p^{te}. \end{aligned} \tag{4}$$

In order to integrate the concept of Siamese templates, we add a target template to enhance feature perception ability. The difference is that our target template is limited to the target region and is obtained from the feature maps instead of the input template image. This seems to be the integration method of Siamese tracking and discriminative tracking. The training frame already contains a larger background region, and Gaussian labels and *ltrb* regression maps serve as discriminative supervision, while the template token is designed to make feature learning more focused. Concisely, we use a RoIAlign [33] operation to achieve our goal, as follows:

$$e_t = \text{RoIAlign}\left(PE\left(x_{im}\right), [x, y, w, h]\right). \tag{5}$$

3.4 Target Classifier and Bounding Box Regressor

Above, we have introduced the structure of joint feature learning backbone and target model prediction. To simplify, we call the joint feature learning backbone as the joint backbone encoder $\mathcal{B}_{enc}$, which is shown in Figure 5. Through the target model prediction, we obtain the target, template, training and test tokens, a joint backbone encoder is used to realize the joint feature extraction and interaction between these features $[e_f, e_t,\ e_{tr}^i, e_{tr}^d, e_{te}]$, as follows:

$$[e_f^{enc}, e_t^{enc},\ \left[e_{tr}^i, e_{tr}^d\right]^{enc}, e_{te}^{enc}] = \mathcal{B}_{enc}\left([e_f, e_t, e_{tr}^i, e_{tr}^d, e_{te}]\right). \tag{6}$$

Here, e_{tr}^i and e_{tr}^d express the initial training token and dynamic training token separately. Subsequently, e_f^{enc} and e_{te}^{enc} is shared to target classifier and bounding box regressor. The encoded foreground target token e_f^{enc} and the test features e_{te}^{enc} pass through the target classifier module to predict the response map, as follows:

$$\begin{aligned} &\mathcal{R} = f_{te}^{enc} * w,\ f_{te}^{enc} \in \mathbb{R}^{768\times18\times18}\ w \in \mathbb{R}^{768\times1\times1}. \\ &s.t.\ \left[f_{te}^{enc}, w\right] = \gamma\left(\left[e_{te}^{enc}, e_f^{enc}\right]\right) \end{aligned} \tag{7}$$

Here, * denotes the convolution operation. In the bounding box regressor, the foreground target weight w is first preprocessed by the linear layer ζ, and then convoluted with the test features f_{te}^{enc} to obtain a single channel attention map f_{att}, The attention map f_{att} and the test features f_{te}^{enc} is fused by the element-wise product, and next input into Fully Convolutional Network (FCN ϕ) to acquire a *ltrb* feature map, as follows:

$$\begin{aligned} &[l, t, r, b] = \phi\left(f_{te}^{enc} \otimes f_{att}\right). \\ &s.t.\ f_{att} = f_{te}^{enc} * \zeta\left(w\right) \end{aligned} \tag{8}$$

Finally, we select and transform (*i.e.,* $s\tau$) the *ltrb* expression into the target state of the test frame by using the peak coordinate of the response map, as follows:

$$\begin{aligned} &\left[x_{reg}, y_{reg}, w, h\right] = s\tau\left(\left[l, t, r, b\right], \left[x_{cls}, y_{cls}\right]\right). \\ &s.t.\ \ \left[x_{cls}, y_{cls}\right] = \arg\max\left(\mathcal{R}\right) \end{aligned} \tag{9}$$

In the training stage, we employ the target classification loss from DiMP [2] as supervision of the response map, and L1 and GIOU [34] loss from STARK [43] to supervise the bounding box prediction, as follows:

$$\begin{aligned} &\mathcal{L}_{total} = \lambda_{cls}\mathcal{L}_{cls}\left(\bar{y}_c, y_c\right) + \\ &\lambda_{l1}\mathcal{L}_{l1}\left(\bar{l}_r, l_r\right) + \lambda_{giou}\mathcal{L}_{giou}\left(\bar{l}_r, l_r\right). \end{aligned} \tag{10}$$

Here, $\{*\}$ denotes the ground truth and $\{\bar{*}\}$ denotes the predicted result. The generation of y_c and l_r refers to the approach of ToMP[30]. $\mathcal{L}_{cls}$ is the target classification loss, while $\mathcal{L}_{l1}$ and $\mathcal{L}_{giou}$ are the bounding box regression loss. We set $\lambda_{cls} = 100$, $\lambda_{l1} = 5$, $\lambda_{giou} = 2$.

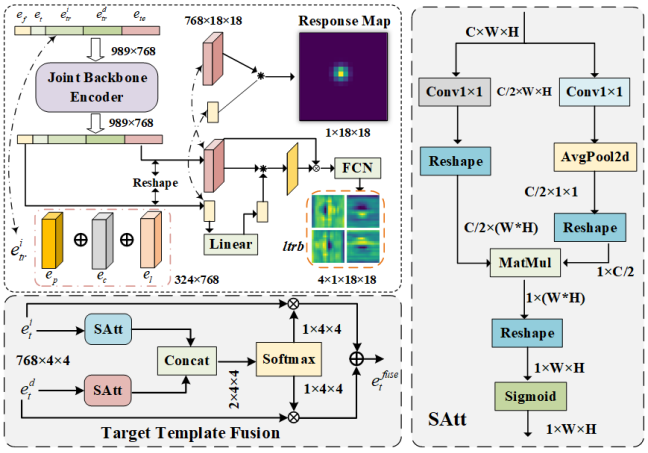


Figure 5: Structure of Target Classifier, Bounding Box Regressor, and Target Template Fusion. The input embedding of the joint backbone encoder is the mixed tokens of foreground target token, target template token, initial training token, dynamic training token, and test token. The content of target template fusion will be introduced in Section 3.5.

3.5 Online Token Update

During online tracking framework, we use annotated initial frame and subsequent predicted frame as our training frames. Especially, the initial frame remains unchanged, and the predicted frame is dynamically changing, the quality of dynamic training frame updates affects the effectiveness of online tracking. As shown in Figure 5, the response map $\mathcal{R}$ can be obtained through the target classifier module. To evade introducing additional branches and two-stage training like MixFormer and STARK [9, 43], refer to the practice of

discrimination tracking [2, 30], and directly determine whether to update the dynamic training token e_{tr}^{d} according to the peak value $\mathcal{R}^{peak}$ of the response map. Normally, the predicted response map approximates the Gaussian classification label. The peak value of the response map not only indicates the location of the tracking, but also indicates the reliability of the position determination. To be concise, we only use the max score of the response map as the updated standard of the dynamic training sample. Since the training sample contains specific bounding box information, it can only be updated by replacement, as follows:

$$\begin{cases} e_{tr} = \left[e_{tr}^{i}, e_{tr}^{i}\right], \ frame = 1, \\ e_{tr} = [e_{tr}^{i}, e_{tr}^{d}], \ \mathcal{R}^{peak} > \theta. \end{cases} \tag{11}$$

Here, θ is an update threshold, which is set 0.95 by default. e_{tr}^{i} denotes the initial frame token, e_{tr}^{d} denotes the dynamic training token when $\mathcal{R}^{peak} > \theta$.

At first, the target template is obtained using a fixed initial training frame. In order to add the target template of dynamic training frames, but not to enhance the length of the mixed tokens, we propose a target template fusion with spatial attention method to achieve this function, which is shown in Figure 5. This template fusion method uses a spatial attention block *SAtt* (as shown at the right sub-figure of Figure 5 to obtain the spatial attention map for adaptive attention-weighted fusion. We perform the element-wise product and addition between the spatial attention map and the initial template token e_{t}^{i}, as well as the dynamic template token e_{t}^{d}, to realize adaptive template fusion. Furthermore, we adopt Softmax function to normalize the attention weight map to make the weight coefficient on the fusion map position meet the sum of 1, as follows:

$$\begin{aligned} (w_i, w_d) &= Soft\max\left(cat\left(SAtt\left(e_t^i\right), SAtt\left(e_t^d\right)\right)\right), \\ e_t^{fusion} &= \left(w_i \otimes e_t^i\right) \oplus \left(w_d \otimes e_t^d\right). \end{aligned} \tag{12}$$

4 Experiments

4.1 Implementation Details

We train our tracker on the training splits of the LaSOT [13], TrackingNet [32], GOT-10k [21], and COCO2017 [28] datasets. We use pre-trained MAE [18] ViT-base and the base version Vim [47] with pre-trained weights with a patch size of 16 to initialize the network parameters of the joint backbone block. We sample 60k subsequences and train for 300 epochs on 4 NVIDIA GeForce RTX 2080ti GPUs 22G. Specifically, our image patch size is 288, the search scale factor is 5, and AdamW [29] optimizer is adopted. The learning rate of AdamW is 1e-4, but that of backbone is 2e-5, and we decay by a factor of 0.2 after 150 and 250 epochs and weight decay of 1e-4. We construct a training sub-sequence by randomly sampling two training frames and a test frame with a 200-frame interval within a video sequence. We then extract the image patches after randomly translating and scaling the image relative to the target bounding box. Moreover, we use random image flipping and color jittering for data augmentation.

To verify the practicability of our proposed one-stream discriminative tracking framework and online tracking method, we show comparative performances on eight challenging benchmarks including LaSOT [13], LaSOText [12], TrackingNet [32], GOT-10k [21], OTB100 [41], NFS [23], UAV123 [31], and TNL2K [38]. Our approach is implemented in Python using PyTorch, and all test experiments are running in the GPU processor of GeForce RTX 4070.

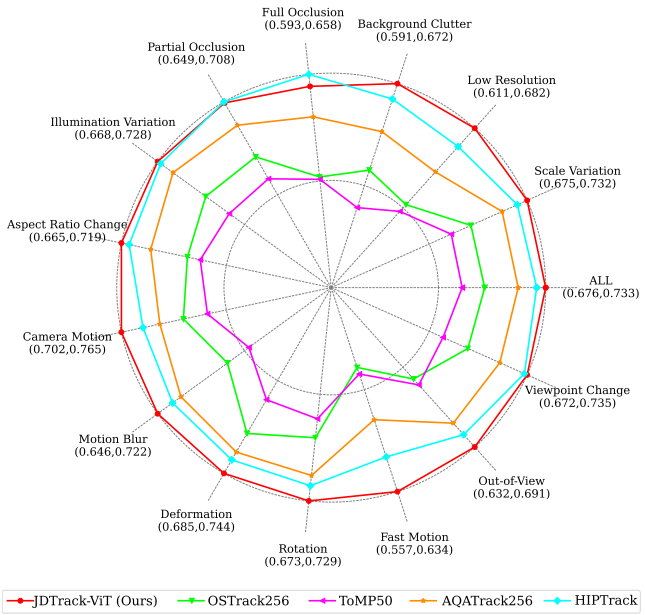


Figure 6: AUC scores of difference challenging attributes on LaSOT[13]. Best viewed in color.

4.2 Comparison to the State of the Art

We exploit a novel one-stream discriminative tracker including a joint feature learning backbone, target model prediction, prediction head, and online update strategy, to construct a high-performance **J**oint **D**iscriminative **Track**ing framework, hence it is called the JDTrack tracker. JDTrack updates the training and template tokens online by default, including JDTrack-ViT and JDTrack-Vim, using ViT-B and Vim-B structure with pre-trained parameters respectively. In this section, we compare our proposed JDTrack-ViT and JDTrack-Vim trackers with the State of the Art (SOTA) trackers on eight challenging tracking benchmarks.

LaSOT [13]: Table 1 shows the comparison results in terms of Precision, Normalized Precision and AUC scores for various trackers. Our proposed one-stream discriminative tracking pipeline and online update modules can enhance the discriminative ability of the target features. Recent DiMP [2], TransT [7], STARK [43], ToMP50 [30], OSTrack [44], MixFormer [9], SwinTrack [27], AQATrack [42], LoRAT [26], and HIPTrack [4] are considered for comparison. Our proposed JDTrack-ViT achieves competitive performance, outperforming ToMP50 with a relative gain of 5.7 %. Overall, the proposed methods have significantly improved tracking performances.

LaSOText [12]: LaSOText is an extended subset of LaSOT that includes 150 additional videos with 15 new categories. As shown in Table 1, Our JDTrack-ViT with 288 image size tracker achieved an AUC score basically similar to that of LoRAT-B [26], and we can observe our JDTrack-ViT outperforms OSTrack and ARTrack by 3.0 % and 4.0% AUC score. Compared with ToMP50, we can see that

Table 1: Analysis of our JDTrack-ViT/Vim compared with SOTA trackers on the LaSOT, LaSOText, TrackingNet, and GOT-10k datasets. Therein, the evaluation metrics include Precision (P), Normalized Precision (NP), Area Under the Curve (AUC) , Success Rate (SR), and Average Overlap (AO) (%). The two best results are highlighted in bold red and blue.

Trackers	Source	LaSOT			LaSOText			TrackingNet			GOT-10k		
		AUC	NP	P	AUC	NP	P	AUC	NP	P	AO	SR0.5	SR0.75
DiMP[2]	ICCV2019	56.9	65.0	56.7	39.2	47.6	45.1	74.0	80.1	68.7	61.1	71.7	49.2
TrDiMP[36]	CVPR2021	63.9	73.0	66.3	-	-	-	78.4	83.3	73.1	67.1	77.7	58.3
TransT[7]	CVPR2021	64.9	73.8	69.0	-	-	-	81.4	86.7	80.3	67.1	76.8	60.9
DTT[45]	ICCV2021	60.1	-	-	-	-	-	79.6	85.0	78.9	68.9	79.8	62.2
STARK[43]	ICCV2021	67.1	76.9	72.2	-	-	-	82.0	86.9	-	68.8	78.1	64.1
ToMP50[30]	CVPR2022	67.6	78.0	72.2	46.7	57.2	53.0	81.2	86.2	78.6	72.0	83.7	66.2
MixFormer1k[8]	CVPR2022	67.9	77.3	73.9	-	-	-	82.6	87.7	81.2	73.2	83.2	70.2
SwinTrack-B[27]	NIPS2022	71.3	-	76.5	49.1	-	55.6	84.0	-	82.8	72.4	80.5	67.8
OSTrack256[44]	ECCV2022	69.1	78.7	75.2	47.4	57.3	53.3	83.1	87.8	82.0	71.0	80.4	68.2
TATrack-B[19]	AAAI2023	69.4	78.2	74.1	-	-	-	83.5	88.3	81.8	77.3	**87.8**	74.1
ROMTrack[5]	ICCV2023	69.3	78.8	75.6	48.9	59.3	55.0	83.6	88.4	82.7	72.9	82.9	70.2
ARTrack256[39]	CVPR2023	70.4	79.5	76.6	46.4		52.3	**84.2**	88.7	83.5	73.5	82.2	70.9
GRM[16]	CVPR2023	69.9	78.0	75.8	-	-	-	84.0	88.7	83.3	73.4	82.9	70.4
UTrack256[14]	ACMMM2023	70.3	80.1	77.1	-	-	-	83.3	**89.3**	**84.3**	75.5	86.4	74.3
MixCVT[9]	TPAMI2024	69.1	78.7	74.7	-	-	-	83.1	88.1	81.6	72.6	82.2	68.8
LoRAT-B[26]	ECCV2024	71.7	80.9	77.3	50.3	61.6	57.1	83.5	87.9	82.1	72.1	81.8	70.7
HIPTrack[4]	CVPR2024	**72.7**	**82.9**	**79.5**	**53.0**	**64.3**	**60.6**	**84.5**	**89.1**	**83.8**	**77.4**	**88.0**	**74.5**
AQATrack256[42]	CVPR2024	71.4	81.9	78.6	**51.2**	**62.2**	**58.9**	83.8	88.6	83.1	73.8	83.2	72.1
JDTrack-Vim	**Ours**	68.2	77.3	73.1	47.3	57.6	53.3	82.8	87.7	81.2	72.8	83.3	69.5
JDTrack-ViT	**Ours**	**73.3**	**82.7**	**79.2**	50.4	60.7	57.5	84.0	87.7	83.2	**78.2**	**87.8**	**78.1**

our JDTrack-ViT/Vim promotes the ToMP50 with a relative AUC gain of 3.7/0.6 (%).

TrackingNet [32]: The experimental results are provided in Table 1. Our JDTrack-ViT/Vim tracker achieves AUC scores of 84.0/82.8 (%), our JDTrack-ViT outperforms the state-of-the-art trackers such as MixFormer, OSTrack, LoRAT-B, AQATrack, etc. Compared with ToMP50, we can see that our JDTrack-ViT/Vim promote the ToMP50 with a relative AUC gain of 2.8/1.6 (%). The competitive results on this dataset further demonstrate the effectiveness of our proposed approach.

GOT-10k [21]: We use the test set of GOT-10k to test our tracker. As shown in Table 1. Our JDTrack-ViT obtains an AO score of 78.2%, surpassing all comparison methods. Moreover, our JDTrack-Vim outperforms the ToMP50 with a relative AO gain of 0.8%. The experiment further demonstrates that our JDTrack is feasible.

TNL2K [38], **UAV123** [31], **NFS** [23], **OTB100** [41]: TNL2K is a recently released large-scale dataset with 700 challenging video sequences. Since there is no original result of ToMP50 on TNL2K dataset, we also test the TNL2K results of ToMP50. As shown in Table 2, our JDTrack-ViT achieves an AUC score of 58.8% on TNL2K dataset, with a relative gain of 4.7%. Moreover, our JDTrack-ViT achieves an AUC score of 71.3/71.0 (%) on OTB100/NFS datasets, which obtains superior performance compared with most SOTA approaches. Especially, our JDTrack-Vim achieves the best AUC results on NFS dataset, surpassing ToMP50 with a 1.4% gain.

Summary: From the comparison of the results in Tables 1 and 2, we have noticed that using ViT to construct a joint feature learning backbone is more effective than Vim. In order to have a clearer visualization of various challenging attributes, we provide radar comparison diagram of each attribute on LaSOT dataset. As in revealed in Figure 6, our JDTrack-ViT tracker has improved all 12 tracking attributes except for partial/full occlusion, which is

Table 2: Comparison results of our JDTrack compared with state-of-the-art trackers on the OTB100, NFS, UAV123, and TNL2K datasets in terms of AUC score.

	OTB100	NFS	UAV123	TNL2K
DiMP	68.4	62.0	65.4	-
TrDiMP	71.1	66.5	67.5	-
TransT	69.4	65.7	69.1	-
STARK	68.1	66.2	68.2	-
ToMP50	70.1	66.9	69.0	54.1
MixFormer1k	-	-	68.7	-
OSTrack256	-	64.7	68.3	54.3
ARTrack256	-	64.3	67.7	57.5
ROMTrack	**71.4**	68.0	-	-
UTrack256	**71.8**	-	-	57.5
MixCVT	70.0	-	70.4	-
AQATrack256	-	-	**70.7**	**57.8**
HIPTrack	71.0	**68.1**	70.5	-
JDTrack-Vim	69.7	**68.3**	69.7	56.0
JDTrack-ViT	71.3	67.3	**71.0**	**58.8**

lower than HIPTrack. This is particularly prominent in scenarios such as fast motion, background clutter, and low resolution. The experimental results from eight popular tracking data show that our proposed tracking framework has strong superiority.

4.3 Ablation Analysis

To demonstrate the effectiveness of our proposed approaches, we perform synthetical ablation studies on all test datasets.

Joint ViT and Vim Block: Our proposed tracking framework is one-stream discriminative tracking pipeline, which is inspired by one-stream Siamese tracking (*e.g.,* OSTrack [44]) and two-stream discriminative tracking methods (*e.g.,* ToMP [30]). Like OSTrack, we use ViT [11] to realize the joint learning of feature extraction and discriminative model prediction. Furthermore, we introduce Vim [47] to implement the joint backbone design. Our baseline method is ToMP, which includes two versions of ToMP using ResNet50 and ResNet101. During actual testing, we found that the ResNet50 has better performance, so we choose ToMP50 as a baseline. We first replace ResNet50 with ViT-base/Vim-base to test ViT/Vim backbone feature extraction ability. As shown in Table 3, the experiment indicates that using only ViT/Vim to extract features is not as good as ResNet50, and the strong feature interaction ability of ViT/Vim cannot be reflected simply as the extraction backbone.

Table 3: Analysis of using ResNet50 and ViT-B/Vim-B backbone and their impacts on the baseline tracker in terms of AUC score.

	ResNet50	ViT-B	Vim-B	OTB100	NFS	LaSOT
ToMP50	√	-	-	70.1	66.9	67.6
	-	√	-	65.8	65.0	66.8
	-	-	√	68.7	65.9	66.1

We note that OSTrack has achieved great success in building a one-stream Siamese tracking architecture of joint feature extraction and fusion. However, the one-stream discriminative tracking method has not been developed. For this reason, we propose a one-stream JDTrack method based on the shortcomings of the two-stream ToMP. OSTrack uses the Siamese paradigm and a non-updated template, while our JDTrack uses the discriminative paradigm and a dynamic training frame and target template. As shown in Table 4, we give the comparison results of ToMP50 and JDTrack using ViT/Vim joint backbones. The experimental results show that we can achieve good performance gain, especially when adopting ViT joint backbone.

Table 4: Comparison results of two-stream ToMP50 and our one-stream JDTrack regarding AUC or AO scores on eight datasets.

	LaSOT	LaSOText	GOT10k	TrackingNet	TNL2K	OTB100	NFS	UAV123
ToMP50	67.6	46.7	72.0	81.2	54.1	70.1	66.9	69.0
JDTrack-Vim	68.2	47.3	72.8	82.8	56.0	69.7	68.3	69.7
JDTrack-ViT	73.3	50.4	78.2	84.0	58.8	71.3	67.3	71.0

Updatable Template Token: Table 5 provides a comparative experiment on whether our JDTrack-ViT tracker uses a target template or not. Although good tracking performance can be achieved without the target template, using the initial target template can obtain relative gains of 1.4/0.6/1.7/0.5/0.2/1.1/0.7/0.6 (%). Moreover, the initial and dynamic template fusion method proposed in Figure 5 and Equation 12 will have a more significant effect, which is also due to the fact that the peak value of the response map predicted by the proposed tracker can more accurately select reliable dynamic training frames and ensure the acquisition of dynamic target templates. Ultimately, the updated target template can better enhance the generation of target-aware features.

Table 5: Analysis of our JDTrack-ViT without (w/o) and with (w/) initial template/dynamic template token, and their impacts on the tracking performance in terms of AUC or AO scores.

JDTrack	LaSOT	LaSOText	GOT10k	TrackingNet	TNL2K	OTB100	NFS	UAV123
w/o	71.2	48.5	74.8	83.2	57.0	69.6	67.2	68.9
w/ e_t^i	72.6	49.1	76.5	83.7	57.2	70.7	67.9	69.5
w/e_t^d	73.3	50.4	78.2	84.0	58.8	71.3	67.3	71.0

Flops, Parameters and Tracking Speed: Above, we have shown that our proposed methods are feasible and effective from substantial ablation experiments. Next, we will further elaborate on network parameters and tracking speed in Table 6. Our methods are improved based on ToMP50, which uses ResNet50 as the feature backbone. The parameter of ResNet50 is 25.56M, but actual trackers only use the network before layer3 (*i.e.,* 8.54M), and the trainable layer3 is 7.1 M. Our one-stream JDTrack structure is relatively concise, only including the joint feature learning backbone and the prediction head. Although the parameter count and computational flops of JDTrack-Vim are very low, the actual tracking speed is similar to JDTrack-ViT. However, the GPU resources required for training are indeed much less, but the tracking performance is relatively poor. Moreover, our tracker's tracking speed may be lower than ToMP50, but the performance gain obtained is still considerable. On the whole, the tracking speed on the GPU processor of GeForce RTX 4070 achieves real-time performance.

Table 6: Flops and network parameters of ToMP50 and our JDTrack trackers. Moreover, we also provide the average inference speed of corresponding trackers in LaSOT 280 test datasets.

	Flops(G)↓	Parameters (M)↓	Speed(FPS)↑
ToMP50	25.71	26.11	55.7
JDTrack-Vim	6.30	24.96	37.2
JDTrack-ViT	73.10	108.83	36.8

5 Conclusion

This paper is devoted to enhancing the discriminative ability of the recent DCF-based trackers from the perspective of a one-stream discriminative tracking pipeline. We inventively refactor a target model prediction and propose joint ViT and Vim backbones to integrate feature extraction and discriminative model prediction for exploiting the one-stream discriminative framework. Moreover, we adopt a simple and practical update mechanism for properly replacing the training token with high confidence and propose a spatial attention fusion temporal dynamic template to achieve the combination of spatiotemporal template and discriminative tracking architecture. Our methods achieve state-of-the-art performance on eight benchmarks, showing the potential ability of our approaches.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62376252, 62473338), Zhejiang Provincial Natural Science Foundation of China (Grant No. LQN25F030016, LZ22F030003), Zhejiang Province Leading Geese Plan (Grant No. 2025C02025, 2025C01056), Jinhua Science and Technology Bureau (Grant No. 2024-4-006).

References

[1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. 2016. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision Workshops*. 850–865.

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. 2019. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6182–6191.

[3] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. 2010. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2544–2550.

[4] Wenrui Cai, Qingjie Liu, and Yunhong Wang. 2024. HIPTrack: Visual Tracking with Historical Prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19258–19267.

[5] Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. 2023. Robust object modeling for visual tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9589–9600.

[6] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. 2022. Backbone is all your need: A simplified architecture for visual object tracking. In *European Conference on Computer Vision*. Springer, 375–392.

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. 2021. Transformer tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8126–8135.

[8] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. 2022. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13608–13618.

[9] Yutao Cui, Cheng Jiang, Gangshan Wu, and Limin Wang. 2024. MixFormer: End-to-End Tracking With Iterative Mixed Attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 6 (2024), 4129–4146.

[10] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. 2017. Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6638–6646.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*, 1–21.

[12] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, et al. 2021. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision* 129 (2021), 439–461.

[13] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. 2019. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5374–5383.

[14] Jie Gao, Bineng Zhong, and Yan Chen. 2023. Unambiguous object tracking by exploiting target cues. In *Proceedings of the 31st ACM international conference on multimedia*. 1997–2005.

[15] Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. 2022. Aiatrack: Attention in attention for transformer visual tracking. In *European Conference on Computer Vision*. Springer, 146–164.

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

[17] Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* (2023).

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16000–16009.

[19] Kaijie He, Canlong Zhang, Sheng Xie, Zhixin Li, and Zhiwen Wang. 2023. Target-aware tracking with long-term context attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 773–780.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[21] Lianghua Huang, Xin Zhao, and Kaiqi Huang. 2019. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence* 43, 5 (2019), 1562–1577.

[22] Sajid Javed, Martin Danelljan, Fahad Shahbaz Khan, Muhammad Haris Khan, Michael Felsberg, and Jiri Matas. 2023. Visual Object Tracking With Discriminative Filters and Siamese Networks: A Survey and Outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 5 (2023), 6552–6574.

[23] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. 2017. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE international conference on computer vision*. 1125–1134.

[24] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. 2018. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8971–8980.

[25] Xiaohai Li, Bineng Zhong, Qihua Liang, Guorong Li, Zhiyi Mo, and Shuxiang Song. 2025. MambaLCT: Boosting Tracking via Long-term Context State Space Model. In *Proceedings of the AAAI conference on artificial intelligence*.

[26] Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. 2024. Tracking meets lora: Faster training, larger model, stronger performance. In *European Conference on Computer Vision*. 300–318.

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

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*. 740–755.

[29] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *International Conference on Learning Representations*.

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

[31] Matthias Mueller, Neil Smith, and Bernard Ghanem. 2016. A benchmark and simulator for UAV tracking. In *European Conference on Computer Vision*. Springer, 445–461.

[32] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. 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.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* 39, 6 (2016), 1137–1149.

[34] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 658–666.

[35] Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. 2022. Transformer tracking with cyclic shifting window attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8791–8800.

[36] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. 2021. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1571–1580.

[37] Qingwang Wang, Liyao Zhou, Pengcheng Jin, Xin Qu, Hangwei Zhong, Haochen Song, and Tao Shen. 2024. TrackingMamba: Visual State Space Model for Object Tracking. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 17 (2024), 16744–16754.

[38] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. 2021. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13763–13773.

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

[40] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 22–31.

[41] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. 2015. Object Tracking Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 9 (2015), 1834–1848.

[42] Jinxia Xie, Bineng Zhong, Zhiyi Mo, Shengping Zhang, Liangtao Shi, Shuxiang Song, and Rongrong Ji. 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.

[43] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. 2021. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10448–10457.

[44] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. 2022. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*. Springer, 341–357.

[45] Bin Yu, Ming Tang, Linyu Zheng, Guibo Zhu, Jinqiao Wang, Hao Feng, Xuetao Feng, and Hanqing Lu. 2021. High-performance discriminative tracking with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9856–9865.

[46] Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. 2024. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 7588–7596.

[47] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. 2024. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. In *Forty-first International Conference on Machine Learning*. 62429–62442.