

# HIPTrack: Visual Tracking with Historical Prompts

Wenrui Cai<sup>1</sup>, Qingjie Liu<sup>1,2,3,\*</sup>, Yunhong Wang<sup>1,3</sup>

<sup>1</sup>State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China

<sup>2</sup>Zhongguancun Laboratory, Beijing, China

<sup>3</sup>Hangzhou Innovation Institute, Beihang University, Hangzhou, China

{wenrui.cai, qingjie.liu, yhwang}@buaa.edu.cn

## Abstract

Trackers that follow Siamese paradigm utilize similarity matching between template and search region features for tracking. Many methods have been explored to enhance tracking performance by incorporating tracking history to better handle scenarios involving target appearance variations such as deformation and occlusion. However, the utilization of historical information in existing methods is insufficient and incomprehensive, which typically requires repetitive training and introduces a large amount of computation. In this paper, we show that by providing a tracker that follows Siamese paradigm with precise and updated historical information, a significant performance improvement can be achieved with completely unchanged parameters. Based on this, we propose a **historical prompt network** that uses refined historical foreground masks and historical visual features of the target to provide comprehensive and precise prompts for the tracker. We build a novel tracker called **HIPTrack** based on the historical prompt network, which achieves considerable performance improvements without the need to retrain the entire model. We conduct experiments on seven datasets and experimental results demonstrate that our method surpasses the current state-of-the-art trackers on LaSOT, LaSOT<sub>ext</sub>, GOT-10k and NfS. Furthermore, the historical prompt network can seamlessly integrate as a plug-and-play module into existing trackers, providing performance enhancements. The source code is available at <https://github.com/WenRuiCai/HIPTrack>.

## 1. Introduction

Visual tracking aims to estimate the position of a target in subsequent frames of a video, given the initial state in the first frame [1, 5, 24, 46]. It can be challenging when the target experiences deformation, scale variation, and partial occlusion over time. While Transformer-based one-stream

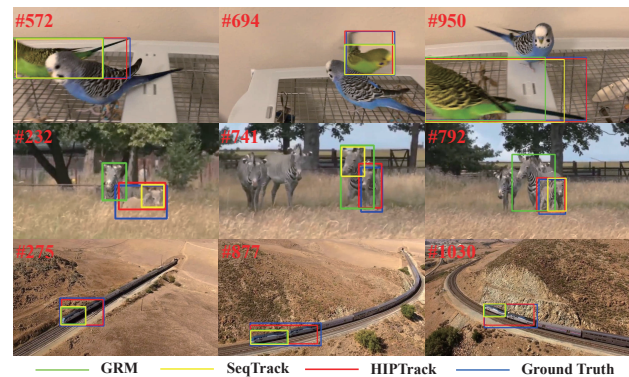


Figure 1. Visualized comparisons of our approach and other excellent trackers GRM [16] and SeqTrack [6]. Our method performs better when the target suffer from occlusion, deformation and scale variation.

trackers [4, 8, 16, 48] have significantly improved the tracking accuracy by incorporating feature extraction and interaction of template and search regions into a single Transformer, they still follow Siamese paradigm that performs similarity matching between template and search regions. Only using the template to predict the target in subsequent frames may not be effective in handling complex scenarios.

There have been numerous attempts to incorporate historical context into trackers using techniques such as space-time memory networks [13], utilizing a tracked frame as auxiliary template [8, 47] and incorporating multiple frames as inputs to the backbone [18, 38]. However, the target position introduced by space-time memory networks lacks accuracy, while the use of two backbones for the current frame and tracked frames introduces a substantial number of parameters. Utilizing a tracked frame as auxiliary template is insufficient and susceptible to introducing distractors. Employing multiple frames as inputs introduces considerable computational complexity while inadequately preserving a substantial amount of historical context. Recently, autoregressive trackers [6, 40] have achieved state-of-the-art performance by tokenizing the target position and incorporat-

\*Corresponding author.

ing a generative decoder to make predictions. Autoregressive trackers can make use of multi-frame historical position information for prediction, and auxiliary template is also introduced in [6]. However, autoregressive trackers still have not fully utilized the historical visual features and require full parameter training with a substantial parameter burden.

In this paper, we conduct an analysis on several trackers [4, 5, 8, 48] that follow Siamese paradigm. We provide updated templates and more accurately cropped search regions while keeping the model unchanged. The results in Figure 2 show significant performance improvements, indicating that high quality historical prompt can improve tracking accuracy without the need for a full parameter retraining. Based on this analysis, we propose **HIPTrack**, a novel tracker that features the **historical prompt network** as its core module. This lightweight module includes an encoder for encoding historical target feature with both positions and visual features of the target, and a decoder for generating historical prompt for current search region.

Compared with [13], HIPTrack incorporates more precise target position information without requiring an additional backbone. The historical prompt encoder introduces precise target position by constructing refined foreground masks, which are then encoded along with the visual features of the target as the historical target features. Compared with [6, 8, 40, 47], HIPTrack introduces more sufficient and comprehensive historical information. We store a large amount of historical target features in the historical prompt decoder and generate prompts tailored to the current search region through adaptive decoding. By leveraging the concept of prompt learning to introduce historical information, HIPTrack achieves significant performance improvements while still maintaining tracking efficiency. Additionally, only the historical prompt network and prediction head need to be trained in HIPTrack, reducing the number of training parameters by more than 80% compared with [40].

We evaluate the performance of our method on 7 datasets and experimental results show that HIPTrack achieves state-of-the-art performance on LaSOT, LaSOT<sub>ext</sub>, GOT-10k and NfS. Figure 1 clearly demonstrates the improved capability of our method in handling scenarios that involve occlusion, deformation and scale variation. The main contributions of this work can be summarized as: **1)** We propose the historical prompt network, a module that encode high quality historical target features and generate effective historical prompts for tracking. **2)** We propose a novel tracker called HIPTrack based on the historical prompt network that eliminates the need for retraining the entire model. **3)** Experimental results show that HIPTrack outperforms all trackers on LaSOT, LaSOT<sub>ext</sub>, GOT-10k, and NfS; Additional experimental results demonstrate that the historical prompt network can serve as a plug-and-play component to improve the performance of current trackers.

## 2. Related Work

### 2.1. Trackers without Historical Information

Trackers that follow Siamese paradigm perform similarity matching between template and search region for tracking [1, 24, 46, 50], which do not leverage historical information. These trackers rely on the predicted bounding box from the previous frame to crop the current search region. Recently, several trackers have employed Transformers to enhance the template-search region interaction, such as TransT [5], DTT [49] and SparseTT [14]. One-stream trackers such as OS-Track [48], SimTrack [4], GRM [16] and DropTrack [42] integrate feature extraction and interaction into one Transformer and significantly boost the tracking performance. However, these methods still do not incorporate any historical information and underperform in complex scenarios such deformation, scale variation and partial occlusion.

### 2.2. Trackers with Historical Information

STMTrack [13] utilizes space-time memory networks [33] to integrate historical information, but its non-shared backbones result in an excessive number of parameters. The template-free design may lead to the neglect of essential information and the target position description is not precise. STARK [47] and MixFormer [8] utilize one search region with high score as an auxiliary template, which is vulnerable to distractors. TATrack [18] and TrDiMP [38] design backbones capable of processing multiple frames to extract historical features, but the computational overhead makes it challenging to retain more historical frames. ARTrack [40] and SeqTrack [6] utilize predicted coordinates of bounding box and employ a generative decoder to integrate historical positions across multiple frames with the current search region feature and make prediction. However, although SeqTrack [6] introduces an auxiliary template, both methods still have not fully exploited the historical visual features of the target. Additionally, autoregressive trackers require full parameter training, ARTrack [40] even requires two-stage training, leading to significant training resource overhead.

### 2.3. Prompt Learning

Prompt learning is a technique used to customize pre-trained models for specific tasks, which has been applied in the fields such as computer vision [20, 39], natural language processing [23, 25], and multimodal studies [21, 34, 35, 51, 52]. Prompt learning typically adjusts the model input or utilizes adapters at various layers to modify the input-output space. In the field of visual tracking, trackers that follow Siamese paradigm can be conceptualized as pre-trained models for feature similarity matching. By incorporating historical information prompts, the similarity matching ability of these trackers can be extended to the temporal dimension.

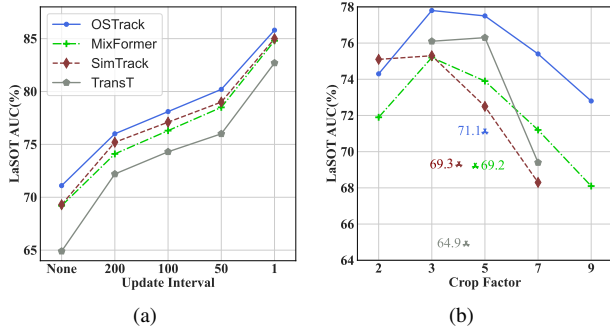


Figure 2. (a) shows the varying performance of trackers on LaSOT [11] as the template update intervals change. (b) shows the varying performance of trackers on LaSOT as the crop factor of current search regions change. A larger crop factor indicates coarser cropping. Each cross symbol represents the baseline of the corresponding color method. Note that TransT [5] does not have a fixed crop factor, so we choose to use an average crop factor instead.

### 3. Method

#### 3.1. Revisit Current Trackers

To explore the performance ceiling of current trackers that follow Siamese paradigm, we conduct two experiments on OSTRack [48], MixFormer [8], SimTrack [4] and TransT [5]. Firstly, we update the template every  $n$  frames, which is cropped with ground truth boxes. Despite a decrease in tracker performance with larger update intervals, as shown in Figure 2(a), when updating the template even with an interval as large as 200, the performance of trackers still remain superior to using only the initial frame as template, which means using more updated target visual features can significantly improve tracking performance. Secondly, we crop the search regions using ground truth boxes in the previous frame instead of the predicted boxes. Although the performance decreases with larger cropping factors, as shown in Figure 2(b), when the cropping factor does not differ much from the baseline, the performance still significantly exceeds the baseline that uses predicted results in the previous frame to crop search regions, which suggests that providing the tracker with more accurate target location information can significantly improve performance. Therefore, for current trackers that follow Siamese paradigm, a substantial performance boost can be attained by using effective historical target feature as prompts.

#### 3.2. Overall Architecture

As shown in Figure 3, we present HIPTrack, which comprises three main components: a feature extraction network, a historical prompt network, and a prediction head network. The feature extraction network extracts the features of the search region interacted with the template, while filtering out background image patches of the search region. The his-

torical prompt network employs a historical prompt encoder to encode the position information and the visual features of the target from the current frame as historical target feature, and appends it to the memory bank in historical prompt decoder. In subsequent tracking, the historical prompt decoder generates historical prompt for each search region and concatenates historical prompt with compressed search region feature along the channel dimension. We adopt the same prediction head structure as OSTRack [48]. Due to the increase in the number of channels after incorporating the historical prompt, we introduce a residual convolutional layer at the input of the prediction head to reduce the channels.

#### 3.3. Feature Extraction Network

The feature extraction network utilizes a Vision Transformer (ViT) [10] as the visual backbone, initialized with the weights of existing trackers and its parameters are entirely frozen. The feature extraction process employs a one-stream approach. Within ViT, similar to OSTRack [48], we incorporate an early Candidate Elimination (CE) module with the same elimination ratio. The CE module is embedded within the attention layers of ViT and filters out search region background tokens with the lowest attention scores in relation to the template token. Our method utilizes the image patches filtered by CE module to construct CE mask as one input of the historical prompt encoder.

#### 3.4. Historical Prompt Network

The historical prompt network is the core module of our method, which consists of an encoder and a decoder, as shown in Figure 4. Inspired by video object segmentation methods [7, 33], we choose to use masks to describe the target position information. The predicted bounding boxes are utilized to generate masks, which are then refined by CE masks to obtain more accurate historical foreground masks of the target. The encoder encodes the refined foreground masks and the target visual features together into the historical target features that serves as historical prompt value and appends it to the memory bank in the decoder, with the compressed search region feature as the key. The decoder utilizes the current search region feature a query and employs attention based on the Euclidean distance for adaptive aggregation of the historical target features to generate historical prompt for the current frame.

##### 3.4.1 Historical Prompt Encoder

As shown in Figure 4, we use the eliminated patches from the feature extraction network to construct a CE mask and generate a bounding box mask using the predicted bounding box. The two masks are combined using a bitwise AND operation to obtain the final refined foreground mask. The refined foreground mask, search region image, and search

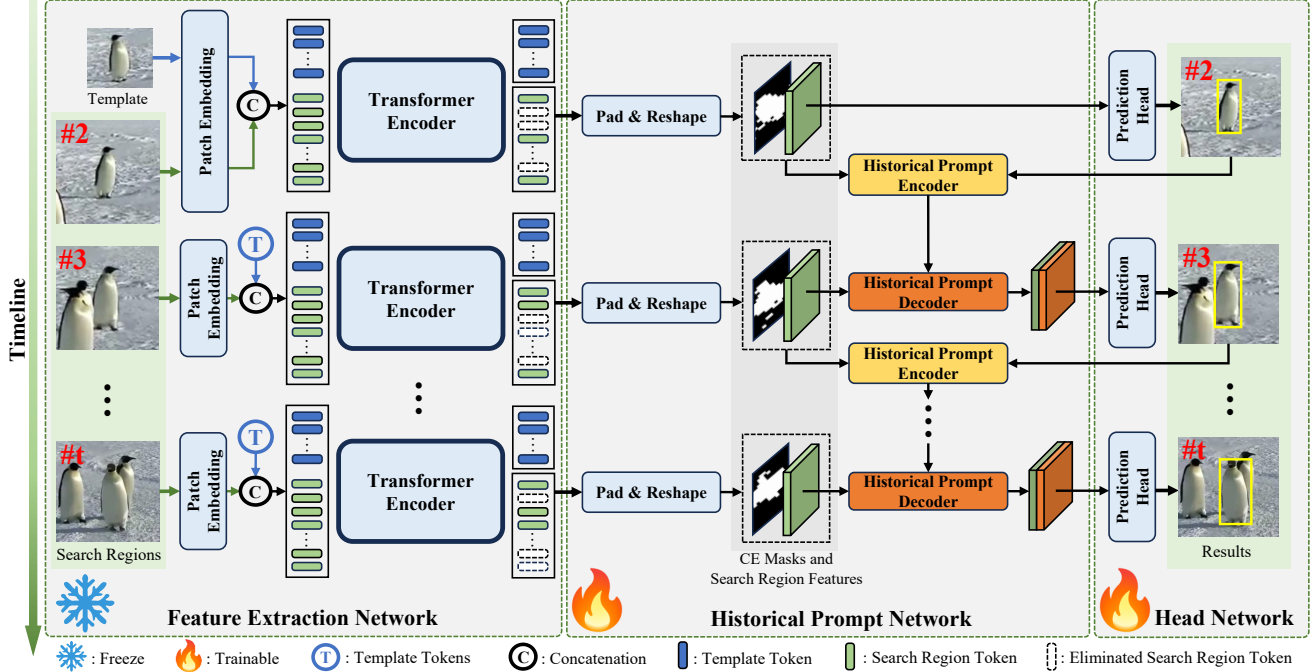


Figure 3. Overview of our proposed HIPTTrack. The whole structure consists of a feature extraction network, a history prompt network, and a head prediction network. The historical prompt network comprises a historical prompt encoder and a historical prompt decoder.

region feature are collectively utilized to encode the historical target feature, also referred to as prompt value.

Initially, the historical prompt encoder concatenates the input search region image with foreground mask along the channel dimension to create a new 4-channel image and feeds the image into a lightweight encoder represented as  $\Phi$ , yielding a feature map  $\mathbf{K} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_K}$  that contains target position information. In our approach,  $\Phi$  utilizes the first three stages of ResNet-18 [17] to ensure that the down-sampling factor aligns with that of the ViT backbone. Besides, we modify the input channel of the first convolutional layer of  $\Phi$  to be 4 while keeping other channels unchanged.

After obtaining the feature  $\mathbf{K}$  that contains the target position information, the historical prompt encoder fuses  $\mathbf{K}$  with the search region feature  $\mathbf{F} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_F}$ . We concatenate  $\mathbf{K}$  and  $\mathbf{F}$  along the channel dimension and feed the result into a convolutional block with residual connections, yielding a fused feature map  $\mathbf{F}'_1 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_P}$ . We denote the residual block as  $f_{RB1}$ , which consists of two  $3 \times 3$  convolutional layers with a padding of 1 and a stride of 1, the first convolutional layer has an output channel count of  $C_P$  and the second convolutional layer maintains the same input and output channels. Later, we adopt a Convolutional Block Attention Module [41] for feature enhancement of  $\mathbf{F}'_1$ , which can be formulated as follows:

$$\begin{aligned}
 W_C &= \text{MLP}(\text{Pool}_{max}^{hw}(\mathbf{F}'_1)) + \text{MLP}(\text{Pool}_{avg}^{hw}(\mathbf{F}'_1)) \\
 W_S &= \text{Conv}([\text{Pool}_{max}^c(\mathbf{F}'_1); \text{Pool}_{avg}^c(\mathbf{F}'_1)]) \\
 \mathbf{F}'_2 &= (\sigma(W_C) \otimes \mathbf{F}'_1) \odot \sigma(W_S)
 \end{aligned} \quad (1)$$

where  $\text{Pool}_{max}^{hw}$  and  $\text{Pool}_{avg}^{hw}$  represent max pooling and average pooling operations applied along the spatial pixel dimension, respectively.  $\text{Pool}_{max}^c$  and  $\text{Pool}_{avg}^c$  represent max pooling and average pooling operations along the channel dimension. The MLP consists of two linear layers and does not alter the dimensions, and  $W_C \in \mathbb{R}^{C_P \times 1}$  is obtained by separately passing two pooling results through the MLP and then adding the two outputs up. Conv is a convolutional layer with a kernel size of 7, padding of 3, and an output channel count of 1.  $W_S \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 1}$  is the output of Conv.  $\sigma$  represents Sigmoid function,  $\otimes$ ,  $\odot$  represent channel-wise multiplication and pixel-wise multiplication, respectively.  $\mathbf{F}'_2 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_P}$  is the result of applying the channel weight and spatial weight to  $\mathbf{F}'_1$ . We further add  $\mathbf{F}'_1$  with  $\mathbf{F}'_2$  and feed the added result into another residual block  $f_{RB2}$  to obtain the final output  $\mathbf{P} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_P}$  as the encoded prompt value.  $f_{RB2}$  maintains the same structural configuration as  $f_{RB1}$ , with the distinction that  $f_{RB2}$  does not alter the number of channels.

### 3.4.2 Historical Prompt Decoder

As shown in Figure 4, the historical prompt decoder stores the historical prompt values in the form of key-value pairs in the memory bank, utilizes the current search region feature to retrieve historical prompt values, and adaptively aggregates the prompt values to generate customized historical prompt relevant to the current search region.

**Memory Bank.** We use compressed search region fea-

tures from the feature extraction network as the prompt key to each prompt value. To reduce computational cost, a  $1 \times 1$  convolutional layer  $\text{Conv}_{key}$  is utilized to reduce the channel dimension of  $F$  from  $C_F$  to  $C_{P_k}$ , obtaining the prompt key that is denoted as  $P_{key} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_{P_k}}$ . After obtaining the prompt values and prompt keys for all the  $\frac{H}{16} \times \frac{W}{16}$  positions in the search region of a specific frame, the historical prompt decoder will flatten them along the spatial dimension and incorporate them into the memory bank altogether, which means that the memory bank will add  $\frac{HW}{16^2}$  prompt key-value pairs. The memory bank retains at most  $T$  tracked frames and is updated every  $\tau$  frames, using a first-in-first-out (FIFO) strategy.

**Decoding.** The decoding process is to adaptively aggregate historical prompt values from the memory bank based on the current search region feature and generate the historical prompt for target prediction. Given the current search region feature  $F$ , to ensure that the query aligns precisely with the prompt keys in the memory bank and to reduce computational cost, we also utilize  $\text{Conv}_{key}$  to reduce the dimension of  $F$  to  $C_{P_k}$  as the query of current frame, which is denoted as  $Q \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_{P_k}}$ . If the prediction of current frame needs to be encoded as historical target feature and added to the memory bank as historical prompt value,  $Q$  can be directly employed as the corresponding prompt key without redundant calculations. Assuming there are a total of  $N$  key-value pairs in the memory bank, the process that the decoder generates historical prompt for current search region can be formulated as follows:

$$\begin{aligned}
 S_{i,j} &= -\|P'_{key_i} - Q_j\|_2^2 \\
 A_{i,j} &= \frac{e^{S_{i,j}}}{\sum_{n=1}^N (e^{S_{n,j}})} \\
 O &= A^T \cdot P'
 \end{aligned} \tag{2}$$

where  $P'_{key} \in \mathbb{R}^{N \times C_{P_k}}$  represents all prompt keys in the memory bank and  $S_{i,j} \in \mathbb{R}^1$  represents the similarity score between the  $i^{th}$  prompt key and the  $j^{th}$  query, which is calculated using the negative of L2 distance.  $A \in \mathbb{R}^{N \times \frac{HW}{16^2}}$  represents the final normalized attention score matrix,  $P' \in \mathbb{R}^{N \times C_P}$  represents all the prompt values in the memory bank, and  $O \in \mathbb{R}^{\frac{HW}{16^2} \times C_P}$  denotes the historical prompt tailored to the current search region, obtained by weighted aggregation based on matrix  $A$ . After reshaping,  $O' \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_P}$  is concatenated with the compressed search region features and fed into the prediction head.

## 4. Experiments

### 4.1. Implementation Details

**Model settings.** We employ the base version of Vision Transformer (ViT-B) [10] in feature extraction network and initialize it using the weights from DropTrack [42]. No-

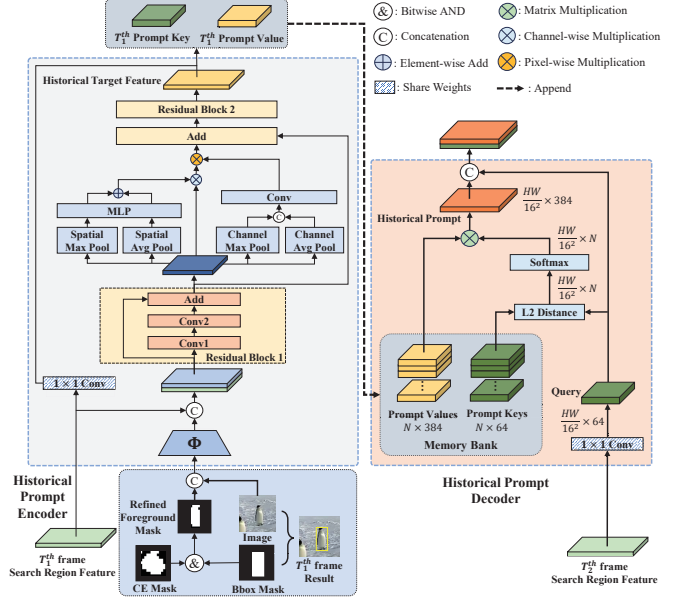


Figure 4. The structure of the historical prompt network that consists of historical prompt encoder and historical prompt decoder.

tably, when evaluating on GOT-10k [19], our initialization weights are trained only on GOT-10k. Throughout the training process, the feature extraction network remains frozen. HIPTrack utilizes a template size of  $192 \times 192$  and a search region size of  $384 \times 384$ . The cropping factors for the template and search region are 2 and 5, respectively. The feature extraction network outputs features with a dimension of  $C_F$  that is set to 768. In historical prompt network, we set the values of  $C_K$ ,  $C_P$  and  $C_{P_k}$  to 256, 384 and 64, respectively. As shown in Table 1, our method requires training only a small number of parameters, while also demonstrating significantly reduced computational complexity during the inference process even when the memory bank is fully utilized. Moreover, our method achieves a speed of 45.3 FPS on one single NVIDIA Tesla V100 GPU, exhibiting a more significant advantage over MACs. This is because during actual tracking, the memory bank is not always full and the encoder is occasionally called at update intervals.

Table 1. Comparison of our method with other excellent trackers in terms of total parameters, trainable parameters, computational complexity and speed. The speed of all methods is tested on V100.

Method	Trainable Params(M)	Params(M)	MACs(G)	Speed (FPS)
<b>HIPTrack</b>	<b>34.1</b>	120.4	<b>66.9</b>	<b>45.3</b>
SeqTrack-B <sub>384</sub> [6]	88.1	<b>88.1</b>	147.9	16.8
ARTrack <sub>384</sub> [40]	181.0	181.0	172.1	13.5
TATrack-L[18]	112.8	112.8	162.4	6.6

**Datasets.** Following previous works [6, 8, 40, 48], when

evaluating the trackers on GOT-10k [19], we only use the *train* splits of GOT-10k for training. Otherwise, we utilize the *train* splits of COCO [27], LaSOT [11], GOT-10k [19], and TrackingNet [32]. In each training step, we select six frames from a video. The temporal intervals between adjacent frames are randomly chosen between 1 and 70. Once selected, the six frames are randomly arranged in either chronological or reverse order, with the first frame serving as the template and the remaining five frames serving as the search frames. For non-video datasets like COCO, we duplicate the same image six times.

**Loss Function.** In our implementation, we utilize focal loss [28] for foreground-background classification and then employ GIoU loss [36] and L1 loss for bounding box regression. The weighting coefficients for focal loss, GIoU loss, and L1 loss are set as 1.0, 5.0, and 2.0, respectively.

**Training and Optimization.** Our tracker is implemented using PyTorch 1.10.1. The entire training process is conducted on 4 NVIDIA Tesla V100 GPUs. During training, we set the batch size to 32 and train the model for 100 epochs. In each epoch, we sample 60,000 videos from all the datasets. We use AdamW optimizer with a weight decay of  $10^{-4}$  and an initial learning rate of  $10^{-4}$ . The learning rate is scheduled to decrease to  $10^{-5}$  after 80 epochs.

**Inference.** For the first 10 frames, the historical prompt network utilizes the historical information from the first frame template as memory bank to generate prompts and the historical information of a tracked frame is added to the memory bank every 5 frames. After the initial 10 frames, the historical prompt network utilizes the historical information stored in the memory bank to generate prompts. The update interval of the memory bank  $\tau$  is set to 20 and the memory bank size  $T$  is set to 150.

## 4.2. Comparisons with the State-of-the-Art

LaSOT [11] is a large-scale long-term dataset. Its *test* split consists of 280 sequences, each exceeding 2,500 frames. We evaluate our method on the *test* split of LaSOT, and the results presented in Table 2 show that our approach outperforms current state-of-the-art methods. In Figure 5, we evaluate the performance of our approach across various challenging tracking scenarios. This observation underscores the adaptability and robustness of our approach.

LaSOT<sub>ext</sub> [12] comprises 1,500 video sequences and 15 distinct target categories that have no overlaps with those in the LaSOT [11] dataset. Table 3 demonstrates a significant superiority of our approach over SeqTrack-B<sub>384</sub>, ARTrack<sub>384</sub> and OTrack<sub>384</sub> on LaSOT<sub>ext</sub>.

GOT-10k [19] contains 9,335 sequences for training and 180 sequences for testing. GOT-10k only allows trackers to be trained using the *train* split. We follow this protocol to train our method on the *train* split and test it on the *test* split. Table 2 indicates that our method surpasses all current

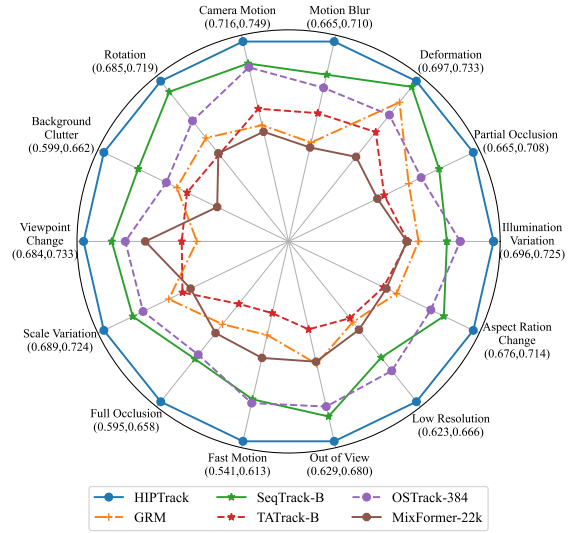


Figure 5. The performance of our method compared with other state-of-the-art trackers in terms of AUC across various scenarios in the LaSOT *test* split.

state-of-the-art methods as well, showcasing the robust capability of our approach in extracting historical information and generating prompts even for unknown categories.

TrackingNet [32] is a large-scale dataset whose *test* split includes 511 sequences covering various object classes and tracking scenes. We report the performance of our method on the *test* split of TrackingNet. Table 2 demonstrates that our method outperforms all current non-autoregressive methods as well as the autoregressive method SeqTrack [6].

UAV123 [31] is a low altitude aerial dataset taken by drones, including 123 sequences, with an average of 915 frames per sequence. The results in Table 4 indicate that our approach rivals to the current state-of-the-art methods on UAV123. The reason for not achieving more significant advancements could be attributed to the relatively smaller target scales in UAV123, which may lead to a decrease in accuracy when using masks to describe the target position.

NfS [22] comprises 100 video sequences, totaling 380,000 video frames. We experiment on the 30FPS version of NfS. The results in Table 4 demonstrate that our approach outperforms all current state-of-the-art approaches.

OTB2015 [43] is a classical testing dataset in visual tracking. It contains 100 short-term tracking sequences covering 11 common challenges, such as target deformation, occlusion and scale variation. The results in Table 4 demonstrate that our approach surpasses current state-of-the-art methods on OTB2015 as well.

## 4.3. Ablation Studies

**Generalization ability of Historical Prompt Network.** To test the general applicability of the historical prompt net-

Table 2. State-of-the-art comparison on LaSOT, GOT-10k and TrackingNet. ‘\*’ denotes for trackers trained only with GOT-10k train split. The best two results are highlighted in **red** and **blue**, respectively.

Method	Source	LaSOT			GOT-10k*			TrackingNet		
		AUC(%)	$P_{Norm}$ (%)	P(%)	AO(%)	$SR_{0.5}$ (%)	$SR_{0.75}$ (%)	AUC(%)	$P_{Norm}$ (%)	P(%)
<b>HIPTrack</b>	<b>Ours</b>	<b>72.7</b>	<b>82.9</b>	<b>79.5</b>	<b>77.4</b>	<b>88.0</b>	<b>74.5</b>	<b>84.5</b>	<b>89.1</b>	<b>83.8</b>
ROMTrack-384 [3]	ICCV23	71.4	81.4	78.2	74.2	84.3	72.4	84.1	<b>89.0</b>	83.7
DropTrack [42]	CVPR23	71.8	<b>81.8</b>	78.1	<b>75.9</b>	<b>86.8</b>	72.0	84.1	88.9	-
ARTrack <sub>384</sub> [40]	CVPR23	<b>72.6</b>	81.7	<b>79.1</b>	75.5	84.3	<b>74.3</b>	<b>85.1</b>	<b>89.1</b>	<b>84.8</b>
SeqTrack-B <sub>384</sub> [6]	CVPR23	71.5	81.1	77.8	74.5	84.3	71.4	83.9	88.8	83.6
GRM [16]	CVPR23	69.9	79.3	75.8	73.4	82.9	70.4	84.0	88.7	83.3
TATrack-B [18]	AAAI23	69.4	78.2	74.1	73.0	83.3	68.5	83.5	88.3	81.8
CTTrack [37]	AAAI23	67.8	77.8	74.0	71.3	80.7	70.3	82.5	87.1	80.3
OTrack <sub>384</sub> [48]	ECCV22	71.1	81.1	77.6	73.7	83.2	70.8	83.9	88.5	83.2
SimTrack [4]	ECCV22	70.5	79.7	-	69.8	78.8	66.0	83.4	87.4	-
MixFormer-22K [8]	CVPR22	69.2	78.7	74.7	70.7	80.0	67.8	83.1	88.1	81.6
SBT [44]	CVPR22	66.7	-	71.7	70.4	80.8	64.7	-	-	-
AiATrack [15]	ECCV22	69.0	79.4	73.8	69.6	80.0	63.2	82.7	87.8	80.4
SwinTrack [26]	NIPS22	71.3	-	76.5	72.4	-	67.8	84.0	-	82.8
SparseTT [14]	IJCAI22	66.0	74.8	70.1	69.3	79.1	63.8	81.7	86.6	79.5
STARK [47]	ICCV21	67.1	77.0	-	68.8	78.1	64.1	82.0	86.9	-

Table 3. The performance of our method and other state-of-the-art trackers on LaSOT<sub>ext</sub>. The best two results are highlighted in **red** and **blue**.

Method	AUC(%)	$P_{Norm}$ (%)	P(%)
<b>HIPTrack</b>	<b>53.0</b>	<b>64.3</b>	<b>60.6</b>
ARTrack <sub>384</sub> [40]	<b>51.9</b>	<b>62.0</b>	<b>58.5</b>
SeqTrack-B <sub>384</sub> [6]	50.5	61.6	57.5
OTrack <sub>384</sub> [48]	50.5	61.3	57.6
AiATrack [15]	47.7	55.6	55.4
SwinTrack [26]	49.1	-	55.6
ToMP [30]	45.9	-	-
KeepTrack [29]	48.2	-	-
LTMU [9]	41.4	49.9	47.3
DiMP [2]	39.2	47.6	45.1

Table 4. The performance of our method and other state-of-the-art trackers on UAV123, NfS and OTB2015 in terms of AUC metrics. The best two results are highlighted in **red** and **blue**.

Method	UAV123	NfS	OTB2015
<b>HIPTrack</b>	<b>70.5</b>	<b>68.1</b>	<b>71.0</b>
ARTrack <sub>384</sub> [40]	<b>70.5</b>	66.8	-
AiATrack [15]	<b>70.6</b>	<b>67.9</b>	69.6
CTTrack-B [37]	68.8	-	-
SeqTrack-B <sub>384</sub> [6]	68.6	66.7	-
DropTrack [42]	-	-	69.6
MixFormer-L [8]	69.5	-	-
KeepTrack [29]	69.7	66.4	<b>70.9</b>
STARK [47]	69.1	-	68.5

work, we integrate it into Transformer-based one-stream trackers DropTrack [42] and OTrack [45], as well as

SiamFC++ [46] that employs an explicit Siamese structure. Due to the limitations of convolutional network-based SiamFC++ in performing candidate elimination, the historical prompt network in SiamFC++ is exclusively constructed using historical predicted bounding boxes to form masks without CE masks. The experimental results, as depicted in Table 5, demonstrate that our proposed historical prompt network significantly improves the performance of existing methods that follow Siamese paradigm. The historical prompt network can still have a great impact on SiamFC++ without introducing precise position information, possibly due to the limited ability of convolutional network-based trackers to comprehensively represent targets. Through the use of richer historical information, the historical prompt network can help refine the representation of the target.

Table 5. A performance comparison of existing trackers and their integration with our proposed historical prompt network on the GOT-10k test set.

Method	AO(%)	$SR_{0.5}$ (%)	$SR_{0.75}$ (%)
DropTrack	75.9	86.8	72.0
DropTrack w/ HIP	<b>77.4</b>	<b>88.0</b>	<b>74.5</b>
OTrack	73.7	83.2	70.8
OTrack w/ HIP	<b>75.4</b>	<b>85.0</b>	<b>73.7</b>
SiamFC++	59.5	69.5	47.9
SiamFC++ w/ HIP	<b>61.0</b>	<b>71.5</b>	<b>49.6</b>

**The Number of sampled search frames.** Training HIPTrack involves sampling multiple frames as search frames from a video. However, due to resource limitations, it is not feasible to sample an excessive number of search frames at

once. Hence, we conducted an analysis on the impact of the number of sampled search frames on the final tracking performance. As depicted in Table 6, the tracker achieves optimal performance with a sample size of 5 frames. A larger number of sampled frames may require a larger model size and training epochs to match, and the scaling capability of the historical prompt network remains to be tested.

Table 6. The performance of our method on the *test* split of LaSOT when setting different number of sampled search frames.

Number	2	3	4	5	6
AUC(%)	72.1	72.5	72.4	<b>72.7</b>	72.4
P <sub>Norm</sub> (%)	82.3	82.7	82.5	<b>82.9</b>	82.6
P(%)	78.9	79.2	79.1	<b>79.5</b>	79.2

**Ablation Studies on Historical Prompt Network.** Historical Prompt Network simultaneously integrates historical positional information and historical visual feature of the target. In order to assess the individual impacts of these two categories of information on HIPTrack, we conducted separate ablation experiments by removing historical refined foreground masks and historical search region features as inputs to the historical prompt network. The results in the first, second, and sixth rows of Table 7 suggest that incorporating both refined foreground masks and search region features are beneficial for tracking.

We also examine the importance of CE mask that describes more precise historical positional information. The results in the third and sixth rows of Table 7 reveal that using candidate elimination to filter background patches to create a refined mask for introducing accurate positional information is essential. Moreover, we explore the necessity of including background search region image patches in the memory bank. The fourth and sixth rows of Table 7 indicate that including all feature maps yields better results, likely because the constructed mask may not completely cover the target region. In fifth row, we replace the L2 distance of attention calculation with dot product, and the comparison with sixth row shows that L2 distance is better to calculate attention score because our query and key reside in exactly the same feature space.

#### 4.4. Qualitative Study

In Figure 6, we visualize the comparative tracking results of different methods after prolonged tracking, considering changes in the target due to occlusion, deformation and scale variation. We also visualize the attention maps in memory bank. The first column represents the template of the first frame in video, the second column shows the comparison of tracking results from different methods, and the third column presents the visualization of attention maps. Figure 6 demonstrates that our method can effectively query and aggregate historical feature of the target during the

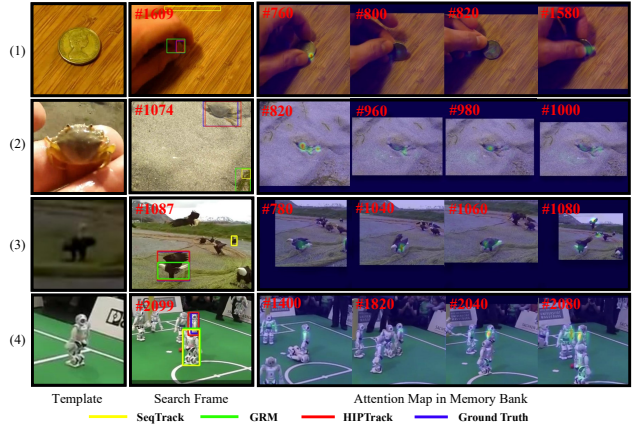


Figure 6. Visualization results of memory bank attention maps after prolonged tracking. Zoom in for a clearer view.

Table 7. Ablation studies on refined foreground masks, search region features, candidate elimination, incorporating background region features into the memory bank and L2 attention.

#	Mask	Feature	CE	Background	L2	LaSOT AUC(%)
1	✓	✗	✓	✓	✓	71.2
2	✗	✓	✗	✓	✓	71.5
3	✓	✓	✗	✓	✓	72.1
4	✓	✓	✓	✗	✓	72.3
5	✓	✓	✓	✓	✗	72.3
6	✓	✓	✓	✓	✓	<b>72.7</b>

tracking process. Compared with other methods like GRM [16] that relies solely on the first frame template, and the autoregressive method SeqTrack [6] that does not fully utilize historical visual features of the target, our approach achieves more accurate tracking results.

## 5. Conclusion

In this work, we identify that trackers that follow Siamese paradigm exhibit significant performance improvements when provided with precise and updated historical information. Based on this observation, we propose the historical prompt network that effectively leverages the visual features and positional information of tracked frames to encode the historical target feature, while adaptively generates historical prompts for subsequent frames to enhance tracking accuracy. Our proposed HIPTrack, which features the historical prompt network as its core module, achieves state-of-the-art performance with a small number of parameters that need to be trained. The historical prompt network can also serve as a plug-and-play component to improve the performance of current trackers.

**Acknowledgements.** This paper was supported by National Natural Science Foundation of China under grants U20B2069 and 62176017.



## References

- [1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865, 2016. 1, 2
- [2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pages 6182–6191, 2019. 7
- [3] Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9589–9600, 2023. 7
- [4] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qihong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: a simplified architecture for visual object tracking. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 375–392. Springer, 2022. 1, 2, 3, 7
- [5] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, pages 8126–8135, 2021. 1, 2, 3
- [6] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14572–14581, 2023. 1, 2, 5, 6, 7, 8
- [7] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021. 3
- [8] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13608–13618, 2022. 1, 2, 3, 5, 7
- [9] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In *CVPR*, pages 6298–6307, 2020. 7
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3, 5
- [11] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, pages 5374–5383, 2019. 3, 6
- [12] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Mingzhen Huang, Juehuan Liu, Yong Xu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129: 439–461, 2021. 6
- [13] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, pages 13774–13783, 2021. 1, 2
- [14] Zhihong Fu, Zehua Fu, Qingjie Liu, Wenrui Cai, and Yunhong Wang. Sparsett: Visual tracking with sparse transformers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 905–912, 2022. 2, 7
- [15] Shenyan Gao, Chunlun Zhou, Chao Ma, Xinggong Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 146–164. Springer, 2022. 7
- [16] Shenyan Gao, Chunlun Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18686–18695, 2023. 1, 2, 7, 8
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4
- [18] Kaijie He, Canlong Zhang, Sheng Xie, Zhixin Li, and Zhiwen Wang. Target-aware tracking with long-term context attention. 2023. 1, 2, 5, 7
- [19] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 2019. 5, 6
- [20] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 2
- [21] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19113–19122, 2023. 2
- [22] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 6
- [23] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 2
- [24] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4282–4291, 2019. 1, 2
- [25] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021. 2
- [26] Liting Lin, Heng Fan, Zhipeng Zhang, Yong Xu, and Haibin Ling. Swintrack: A simple and strong baseline for transformer tracking. *Advances in Neural Information Processing Systems*, 35:16743–16754, 2022. 7

- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 6
- [28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 6
- [29] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13444–13454, 2021. 7
- [30] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8731–8740, 2022. 7
- [31] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, pages 445–461, 2016. 6
- [32] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, pages 300–317, 2018. 6
- [33] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9226–9235, 2019. 2, 3
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2
- [35] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Densclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18082–18091, 2022. 2
- [36] Hamid Rezaatofghi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019. 6
- [37] Zikai Song, Run Luo, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Compact transformer tracker with correlative masked modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 7
- [38] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, pages 1571–1580, 2021. 1, 2
- [39] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 2
- [40] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9697–9706, 2023. 1, 2, 5, 7
- [41] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 4
- [42] Qiangqiang Wu, Tianyu Yang, Ziquan Liu, Baoyuan Wu, Ying Shan, and Antoni B. Chan. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14561–14571, 2023. 2, 5, 7
- [43] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015. 6
- [44] Fei Xie, Chunyu Wang, Guangting Wang, Yue Cao, Wankou Yang, and Wenjun Zeng. Correlation-aware deep tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8751–8760, 2022. 7
- [45] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Joint group feature selection and discriminative filter learning for robust visual object tracking. In *ICCV*, pages 7950–7960, 2019. 7
- [46] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, pages 12549–12556, 2020. 1, 2, 7
- [47] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10448–10457, 2021. 1, 2, 7
- [48] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*, pages 341–357. Springer, 2022. 1, 2, 3, 5, 7
- [49] Bin Yu, Ming Tang, Linyu Zheng, Guibo Zhu, Jinqiao Wang, Hao Feng, Xuetao Feng, and Hanqing Lu. High-performance discriminative tracking with transformers. In *ICCV*, pages 9856–9865, 2021. 2
- [50] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 2
- [51] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2
- [52] Jiawen Zhu, Simiao Lai, Xin Chen, Dong Wang, and Huchuan Lu. Visual prompt multi-modal tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9516–9526, 2023. 2