MixFormer: End-to-End Tracking with Iterative Mixed Attention

Yutao Cui   Cheng Jiang   Limin Wang ✉   Gangshan Wu

State Key Laboratory for Novel Software Technology, Nanjing University, China

{cuiyutao,mg1933027}@smail.nju.edu.cn   {lmwang,gswu}@nju.edu.cn

Abstract

Tracking often uses a multi-stage pipeline of feature extraction, target information integration, and bounding box estimation. To simplify this pipeline and unify the process of feature extraction and target information integration, we present a compact tracking framework, termed as MixFormer*, built upon transformers. Our core design is to utilize the flexibility of attention operations, and propose a Mixed Attention Module (MAM) for simultaneous feature extraction and target information integration. This synchronous modeling scheme allows to extract target-specific discriminative features and perform extensive communication between target and search area. Based on MAM, we build our MixFormer tracking framework simply by stacking multiple MAMs with progressive patch embedding and placing a localization head on top. In addition, to handle multiple target templates during online tracking, we devise an asymmetric attention scheme in MAM to reduce computational cost, and propose an effective score prediction module to select high-quality templates. Our MixFormer sets a new state-of-the-art performance on five tracking benchmarks, including LaSOT, TrackingNet, VOT2020, GOT-10k, and UAV123. In particular, our MixFormer-L achieves NP score of 79.9% on LaSOT, 88.9% on TrackingNet and EAO of 0.555 on VOT2020. We also perform in-depth ablation studies to demonstrate the effectiveness of simultaneous feature extraction and information integration. Code and trained models are publicly available at* https://github.com/MCG-NJU/MixFormer.

1. Introduction

Visual object tracking [1, 4, 4, 18, 24, 36, 42, 45] has been a fundamental task in computer vision area for decades, aiming to estimate the state of an arbitrary target in video sequences given its initial status. It has been successfully deployed in various applications such as human computer interaction [34] and visual surveillance [55]. However, how
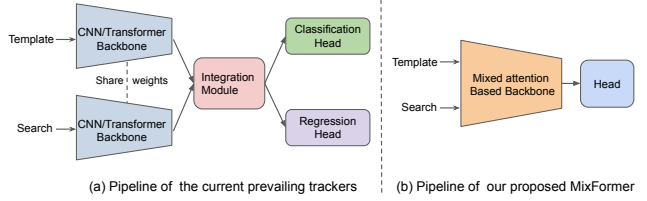

✉ : Corresponding author.

Figure 1. **Comparison of tracking pipeline**. (a) The dominant tracking framework contains three components: a convolutional or transformer backbone, a carefully-designed integration module, and task-specific heads. (b) Our MixFormer is more compact and composed of two components: a target-search mixed attention based backbone and a simple localization head.

to design a simple yet effective end-to-end tracker is still challenging in real-world scenarios. The main challenges are from aspects of scale variations, object deformations, occlusion, and confusion from similar objects.

Current prevailing trackers typically have a multi-stage pipeline as shown in Fig. 1. It contains several components to accomplish the tracking task: (1) a backbone to extract generic features of tracking target and search area, (2) an integration module to allow information communication between tracking target and search area for subsequent target-aware localization, (3) task-specific heads to precisely localize the target and estimate its bounding box. Integration module is the key of tracking algorithms as it is responsible for incorporating the target information to bridge the steps of generic feature extraction and target-aware localization. Traditional integration methods include correlation-based operations (e.g. SiamFC [2], SiamRPN [29], CRPN [18], SiamFC++ [56], SiamBAN [8], OCEAN [64]) and online learning algorithms (e.g., DCF [36], KCF [22], CSR-DCF [37], ATOM [12], DiMP [3], FCOT [9]). Recently, thanks to its global and dynamic modeling capacity, transformers [46] are introduced to perform attention based integration and yields good tracking performance (e.g., TransT [6], TMT [49], STMTrack [19], TREG [10], STARK [57], DTT [59]). However, these transformer based trackers still depend on the CNN for generic feature extraction, and only apply attention operations in the latter high-level and abstract representation space. We analyze

that these CNN representations are limited as they are typically pre-trained for generic object recognition and might neglect finer structure information for tracking. In addition, these CNN representations employ local convolutional kernels and lack global modeling power. Therefore, CNN representation is still their bottleneck, which prevents them from fully unleashing power of self-attention for the whole tracking pipeline.

To overcome the above issue, we present a new perspective on tracking framework design that generic feature extraction and target information integration should be coupled together within a unified framework. This coupled processing paradigm shares several key advantages. First, it will enable our feature extraction to be more specific to the corresponding tracking target and capture more target-specific discriminative features. Second, it also allows the target information to be more extensively integrated into search area, and thereby to better capture their correlation. In addition, this will result in a more compact and neat tracking pipeline only with a single backbone and tracking head, without an explicit integration module.

Based on the above analysis, in this paper, we introduce the *MixFormer*, a simple tracking framework designed for unifying the feature extraction and target integration solely with a transformer-based architecture. Attention module is a very flexible architectural building block with dynamic and global modeling capacity, which makes few assumption about the data structure and could be generally applied for general relation modeling. Our core idea is to utilize this flexibility of attention operation, and present a *mixed attention module* (MAM) that performs both of feature extraction and mutual interaction of target template and search area at the same time. In particular, in our MAM, we devise a hybrid interaction scheme with both self-attention and cross-attention operations on the tokens from target template and search area. The self-attention is responsible to extract their own features of target or search area, while the cross-attention allows for the communications between them to mix the target and search area information. To reduce computational cost of MAM and thereby allow for multiple templates to handle object deformation, we further present a customized *asymmetric* attention scheme by pruning the unnecessary target-to-search area cross-attention.

Following the successful transformer architecture in image recognition, we build our MixFormer backbone by stacking the layers of Patch Embedding and MAM, and finally place a simple localization head to yield our whole tracking framework. As a common practice in dealing with object deformation during tracking procedure, we also propose a score based target template update mechanism and our MixFormer could be easily adapted for multiple target template inputs. Extensive experiments on several benchmarks demonstrate that *MixFormer* sets a new state-of-the-

art performance, with a real-time running speed of 25 FPS on a GTX 1080Ti GPU. Especially, *MixFormer-L* surpasses STARK [57] by 5.0% (EAO score) on VOT2020, 2.9% (NP score) on LaSOT and 2.0% (NP score) on TrackingNet.

The main contributions are summarized as follows:

- We propose a compact end-to-end tracking framework, termed as *MixFormer*, based on iterative Mixed Attention Modules (MAM). It allows for extracting target-specific discriminative features and extensive communication between target and search simultaneously.

- For online template update, we devise a customized asymmetric attention in MAM for high efficiency, and propose an effective score prediction module to select high-quality templates, leading to an efficient and effective online transformer-based tracker.

- The proposed *MixFormer* sets a new state-of-the-art performance on five challenging benchmarks, including VOT2020 [26], LaSOT [17], TrackingNet [41], GOT-10k [23], and UAV123 [40].

2. Related Work

Tracking Paradigm. Current prevailing tracking methods can be summarized as a three-parts architectures, containing (i) a backbone to extract generic features, (ii) an integration module to fuse the target and search region information, (iii) heads to produce the target states. Generally, most trackers [3,7,12,28,47] used ResNet as the backbone. For the most important integration module, researchers explored various methods. Siamese-based trackers [2,21,29, 66] combined a correlation operation with the Siamese network, modeling the global dependencies between the target and search. Some online trackers [3,9,11,12,22,24,30,36] learned an target-dependent model for discriminative tracking. Furthermore, some recent trackers [7,10,19,49,57] introduced a transformer-based integration module to capture more complicated dependencies and achieved impressive performance. Instead, we propose a fully end-to-end transformer tracker, solely containing a MAM based backbone and a simple head, leading to a more accurate tracker with neat and compact architecture.

Vision Transformer. The Vision Transformer (ViT) [15] first presented a pure vision transformer architecture, obtaining an impressive performance on image classification. Some works [32,51,61] introduced design changes to better model local context in vision Transformers. For example, PVT [51] incorporated a multi-stage design (without convolutions) for Transformer similar to multi-scales in CNNs. CVT [52] combined CNNs and Transformers to model both local and global dependencies for image classification in an efficient way. Our MixFomer uses the pre-trained CVT
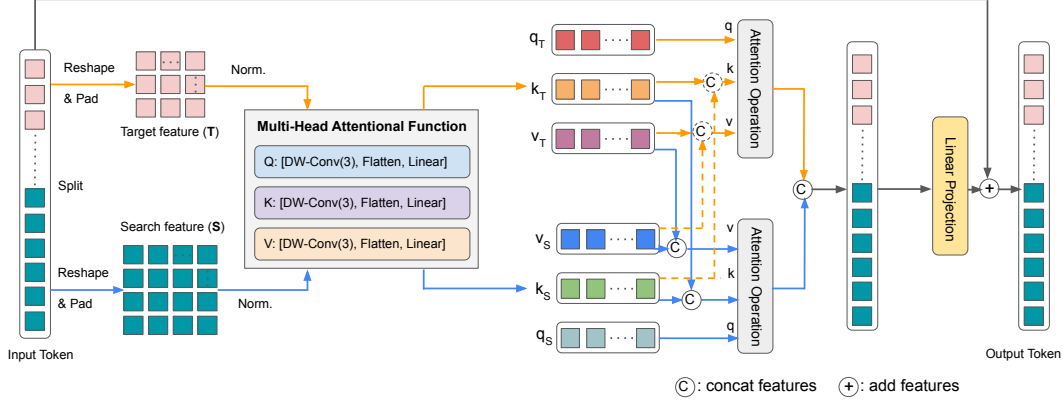
Figure 2. **Mixed Attention Module (MAM)** is a flexible attention operation that unifies the process of feature extraction and information integration for target template and search area. This mixed attention has dual attention operations where self-attention is performed to extract features from itself while cross-attention is conducted to communicate between target and search. This MAM could be easily implemented with a concatenated token sequence. To further improve efficiency, we propose an asymmetric MAM by pruning the target-to-search cross attention (denoted by dashed lines).

models, but there are some fundamental differences. (i) The proposed MAM performs dual attentions for both feature extraction and information integration, while CVT uses self attention to solely extract features. (ii) The learning tasks are different, and the corresponding input and the head are different. We use multiple templates together with the search region as input and employ a corner-based or query-based localization head for bounding box generation, while CVT is designed for image classification. (iii) We further introduce an asymmetric mixed attention and a score prediction module for the specific task of online tracking.

Attention machenism has been also explored in object tracking recently. CGCAD [16] and SiamAttn [60] introduced a correlation-guided attention and self-attention to perform discriminative tracking. TransT [7] designed a transformer-based fusion network for target-search information incorporation. These methods still relied on post-processing for box generation. Inspired by DETR [5], STARK [57] further proposed an end-to-end transformer-based tracker. However, it still followed the paradigm of *Backbone-Integration-Head*, with separate feature extraction and information integration modules. Meanwhile, TREG [10] proposed a target-aware transformer for regression branch and can generate accurate prediction in VOT2021 [27]. Inspired by TREG, we formulate mixed attention mechanism by using both self attention and cross attention. In this way, our MixFormer unifies the two processes of feature extraction and information integration with an iterative MAM based backbone, leading to a more compact, neat and effective end-to-end tracker.

3. Method

In this section, we present our end-to-end tracking framework, termed as *MixFormer*, based on iterative mixed

attention modules (MAM). First, we introduce our proposed MAM to unify the process of feature extraction and target information incorporation. This simultaneous processing scheme will enable our feature extraction to be more specific to the corresponding tracking target. In addition, it also allows the target information integration to be performed more extensively and thus to better capture the correlation between target and search area. Then, we present the whole tracking framework of MixFormer, which only includes a MAM-based backbone and localization head. Finally, we describe the training and inference of MixFormer by devising a confidence score based target template update mechanism to handle object deformation in tracking procedure.

3.1. Mixed Attention Module (MAM)

Mixed attention module (MAM) is the core design to pursue a neat and compact end-to-end tracker. The input to our MAM is the *target template* and *search area*. It aims to simultaneously extract their own long-range features and fuse the interaction information between them. In contrast to the original Multi Head Attention [46], MAM performs dual attention operations on two separate tokens sequences of target template and search area. It carries out self-attention on tokens in each sequence themselves to capture the target or search specific information. Meanwhile, it conducts cross-attention between tokens from two sequences to allow communication between target template and search area. As shown in Fig. 2, this mixed attention mechanism could be implemented efficiently via a concatenated token sequence.

Formally, given a concatenated tokens of multiple targets and search, we first split it into two parts and reshape them to 2D feature maps. In order to achieve additional modeling of local spatial context, a separable depth-wise convolutional projection layer is performed on each feature
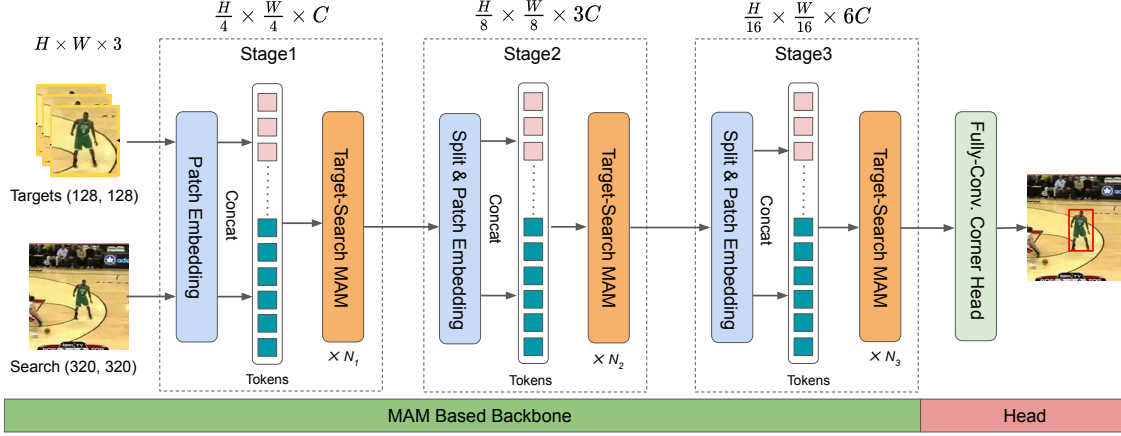
Figure 3. **MixFormer** presents a compact end-to-end framework for tracking without explicitly decoupling steps of feature extraction and target information integration. It is only composed of a single MAM backbone and a localization head.

map (i.e., *query*, *key* and *value*). It also provides efficiency benefits by allowing the down-sampling in *key* and *value* matrices. Then each feature map of target and search is flattened and processed by a linear projection to produce queries, keys and values of the attention operation. We use q_t, k_t and v_t to represent target, q_s, k_s and v_s to represent search region. The mixed attention is defined as:

$$k_m = \text{Concat}(k_t, k_s), \quad v_m = \text{Concat}(v_t, v_s),$$
$$\text{Attention}_t = \text{Softmax}(\frac{q_t k_m^T}{\sqrt{d}})v_m, \qquad (1)$$
$$\text{Attention}_s = \text{Softmax}(\frac{q_s k_m^T}{\sqrt{d}})v_m,$$

where d represents the dimension of the key, Attention_t and Attention_s are the attention maps of the target and search respectively. It contains both self attention and cross attention which unifies the feature extraction and information integration. Finally, the targets token and search token are concatenated and processed by a linear projection.

Asymmetric mixed attention scheme. Intuitively, the cross attention from the targets query to search area is not so important and might bring negative influence due to potential distractors. To reduce computational cost of MAM and thereby allowing for efficiently using multiple templates to deal with object deformation, we further present a customized *asymmetric* mixed attention scheme by pruning the unnecessary target-to-search area cross-attention. This asymmetric mixed attention is defined as follows:

$$\text{Attention}_t = \text{Softmax}(\frac{q_t k_t^T}{\sqrt{d}})v_t,$$
$$\qquad (2)$$
$$\text{Attention}_s = \text{Softmax}(\frac{q_s k_m^T}{\sqrt{d}})v_m.$$

In this manner, the template tokens in each MAM could remain unchanged during tracking process since it avoids influence by the dynamic search regions.

Discussions. To better expound the insight of the mixed attention, we make a comparison with the attention mechanism used by other transformer trackers. Different with our mixed attention, TransT [6] uses ego-context augment and cross-feature augment modules to perform self attention and cross attention progressively in two steps. Compared to the transformer encoder of STARK [57], our MAM shares a similar attention mechanism but with three notable differences. First, we incorporate the spatial structure information with a depth-wise convolution while they use positional encoding. More importantly, our MAM is built as a multi-stage backbone for both feature extraction and information integration, while they depend on a separate CNN backbone for feature extraction and only focus on information integration in a single stage. Finally, we also propose a different asymmetric MAM to further improve the tracking efficiency without much accuracy drop.

3.2. MixFormer for Tracking

Overall Architecture. Based on the MAM blocks, we build the MixFormer, a compact end-to-end tracking framework. The main idea of MixFormer is to progressively extract coupled features for target template and search area, and deeply perform the information integration between them. Basically, it comprises two components: a backbone composed of iterative target-search MAMs, and a simple localization head to produce the target bounding box. Compared with other prevailing trackers by decoupling the steps of feature extraction and information integration, it leads to a more compact and neat tracking pipeline only with a single backbone and tracking head, without an explicit integration module or any post-processing. The overall architecture is depicted in Fig. 3.

MAM Based Backbone. Our goal is to couple both the generic feature extraction and target information integration within a unified transformer-based architecture. The
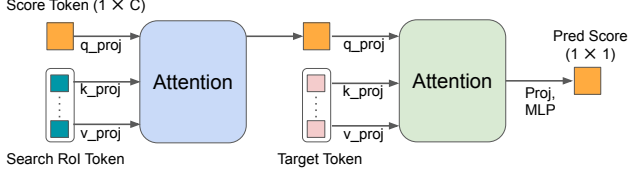
Figure 4. Structure of the Score Prediction Module (SPM).

MAM-based backbone employs a progressive multi-stage architecture design. Each stage is defined by a set of N MAM and MLP layers operating on the same-scaled feature maps with the identical channel number. All stages share the similar architecture, which consists of an overlapped patch embedding layer and N_i target-search mixed attention modules (i.e., a combination of MAM and MLP layers in implementation).

Specifically, given T templates (i.e., the first template and $T-1$ online templates) with the size of $T \times H_t \times W_t \times 3$ and a search region (a cropped region according to the previous target states) with the size of $H_s \times W_s \times 3$, we first map them into overlapped patch embeddings using a *convolutional Token Embedding* layer with stride 4 and kernel size 7. The convolutional token embedding layer is introduced in each stage to grow the channel resolution while reducing the spatial resolution. Then we flatten the patch embeddings and concatenate them, yielding a fused token sequence with the size of $(T \times \frac{H_t}{4} \times \frac{W_t}{4} + \frac{H_s}{4} \times \frac{W_s}{4}) \times C$, where C equals to 64 or 192, H_t and W_t is 128, H_s and W_s is 320 in this work. After that, the concatenated tokens pass through N_i target-search MAM to perform both feature extraction and target information incorporation. Finally, we obtain the token sequence of size $(T \times \frac{H_t}{16} \times \frac{W_t}{16} + \frac{H_s}{16} \times \frac{W_s}{16}) \times 6C$. More details about the MAM backbones could be found in the Section 4.1 and Table 2. Before passed to the prediction head, the search tokens are split and reshaped to the size of $\frac{H_s}{16} \times \frac{W_s}{16} \times 6C$. Particularly, we do not apply the multiscale feature aggregation strategy, commonly used in other trackers (e.g., SiamRPN++ [28], STARK [57]).

Corner Based Localization Head. Inspired by the corner detection head in STARK [57], we employ a fully-convolutional corner based localization head to directly estimate the bounding box of tracked object, solely with several Conv-BN-ReLU layers for the top-left and the bottom-right corners prediction respectively. At last, we can obtain the bounding box by computing the expectation over corner probability distribution [31]. The difference with STARK lies in that ours is a fully convolutional head while STARK highly relies on both encoder and the decoder with more complicated design.

Query Based Localization Head. Inspired by DETR [5], we propose to employ a simple query based localization head. This sparse localization head can verify the generalization ability of our MAM backbone and yield a pure transformer-based tracking framework. Specifically, we add an extra learnable *regression token* to the sequence of the final stage and use this token as an anchor to aggregate information from entire target and search area. Finally, a FFN of three fully connected layers is employed to directly regress the bounding box coordinates. This framework does not use any post-processing technique either.

3.3. Training and Inference

Training. The training process of our MixFormer generally follows the standard training recipe of current trackers [7, 57]. We first pre-train our MAM with a CVT model [52], and then fine-tune the whole tracking framework on the target dataset. Specifically, a combination of L_1 loss and GIoU loss [44] is employed as follows:

$$L_{loc} = \lambda_{L1} L_1(B_i, \hat{B}_i) + \lambda_{giou} L_{giou}(B_i, \hat{B}_i), \quad (3)$$

where $\lambda_{L1} = 5$ and $\lambda_{giou} = 2$ are the weights of the two losses, B_i is the ground-truth bounding box and $\hat{B}_i$ is the predicted bounding box of the targets.

Template Online Update. Online templates play an important role in capturing temporal information and dealing with object deformation and appearance variations. However, it is well recognized that poor-quality templates may lead to inferior tracking performance. As a consequence, we introduce a score prediction module (SPM), described in Fig. 4, to select reliable online templates determined by the predicted confidence score. The SPM is composed of two attention blocks and a three-layer perceptron. First, a learnable *score token* serves as a query to attend the search ROI tokens. It enables the score token to encode the mined target information. Next, the score token attends to all positions of the initial target token to implicitly compare the mined target with the first target. Finally, the score is produced by the MLP layer and a sigmoid activation. The online template is treated as negative when its predicted score is below than 0.5.

For the SPM training, it is performed after the backbone training and we use a standard cross-entropy loss:

$$L_{score} = y_i \log(p_i) + (1 - y_i) \log(1 - p_i), \quad (4)$$

where y_i is the ground-truth label and p_i is the predicted confidence score.

Inference. During inference, multiple templates, including one static template and N dynamic online templates, together with the cropped search region are fed into MixFormer to produce the target bounding box and the confidence score. We update the online templates only when the update interval is reached and select the sample with the highest confidence score.

	KCF [22]	STM [43]	SiamMask [50]	D3S [35]	SuperDiMP [3]	AlphaRef [58]	OceanPlus [62]	RPT [38]	DualTFR [54]	STARK [57]	**MixFormer-1k**	**MixFormer-22k**	**MixFormer-L**
EAO	0.154	0.308	0.321	0.439	0.305	0.482	0.491	0.530	0.528	0.505	0.527	**0.535**	**0.555**
Accuracy	0.407	0.751	0.624	0.699	0.492	0.754	0.685	0.700	0.755	0.759	0.746	**0.761**	**0.762**
Robustness	0.432	0.574	0.648	0.769	0.745	0.777	0.842	**0.869**	0.836	0.817	0.833	0.854	**0.855**

Table 1. State-of-the-art comparison on VOT2020 [26]. The best two results are shown in red and blue fonts. Our trackers use Alpha-Refine [58] to predict masks. MixFormer-1k is pretrained with ImageNet-1k. Others are pretrained with ImageNet-22k.

	Output Size	Layer Name	MixFormer		MixFormer-L	
Stage1	$S: 80 \times 80$, $T: 32 \times 32$	Conv. Embed.	7×7, 64, stride 4		7×7, 192, stride 4	
	$S: 80 \times 80$, $T: 32 \times 32$	MAM MLP	$H_1 = 1$ $D_1 = 64$ $R_1 = 4$	$\times 1$	$H_1 = 3$ $D_1 = 192$ $R_1 = 4$	$\times 2$
Stage2	$S: 40 \times 40$, $T: 16 \times 16$	Conv. Embed.	3×3, 192, stride 2		3×3, 768, stride 2	
	$S: 40 \times 40$, $T: 16 \times 16$	MAM MLP	$H_2 = 3$ $D_2 = 192$ $R_2 = 4$	$\times 4$	$H_2 = 12$, $D_2 = 768$, $R_2 = 4$	$\times 2$
Stage3	$S: 20 \times 20$, $T: 8 \times 8$	Conv. Embed.	3×3, 384, stride 2		3×3, 1024, stride 2	
	$S: 20 \times 20$, $T: 8 \times 8$	MAM MLP	$H_3 = 6$ $D_3 = 384$ $R_3 = 4$	$\times 16$	$H_3 = 16$ $D_3 = 1024$ $R_3 = 4$	$\times 12$
MACs			35.61 M		183.89 M	
FLOPs			23.04 G		127.81 G	
Speed (1080Ti)			25 FPS		18 FPS	

Table 2. MAM based backbone architectures for MixFormer and MixFormer-L. The input is a tuple of templates with shape of $128 \times 128 \times 3$ and search region with shape of $320 \times 320 \times 3$. S and T represent for the search region and template. H_i and D_i is the head number and embedding feature dimension in the i-th stage. R_i is the feature dimension expansion ratio in the MLP layer.

4. Experiments

4.1. Implementation Details

Our trackers are implemented using Python 3.6 and Py-Torch 1.7.1. The MixFormer training is conducted on 8 Tesla V100 GPUs. Especially, MixFormer is a neat tracker **without** post-processing, positional embedding and multi-layer feature aggregation strategy.

Architectures. As shown in Table 2, we instantiate two models, MixFormer and MixFormer-L, with different parameters and FLOPs by varying the number of MAM blocks and the hidden feature dimension in each stage. The backbone of MixFormer and MixFormer-L are initialized with the CVT-21 and CVT24-W [52] (first 16 layers are employed) pretrained on ImageNet [14] respectively.

Training. The training set includes TrackingNet [41], La-SOT [17], GOT-10k [23] and COCO [33] training dataset, which is the same as DiMP [3] and STARK [57]. While for GOT-10k test, we train our tracker by only using the GOT10k train split following its standard protocol. The whole training process of MixFormer consists of two stages, which contains the first 500 epochs for backbones and heads, and extra 40 epochs for score prediction head. We train the MixFormer by using ADAM [25] with weight de-

cay 10^{-4}. The learning rate is initialized as 1e-4 and decreased to 1e-5 at the epoch of 400. The sizes of search images and templates are 320×320 pixels and 128×128 pixels respectively. For data augmentations, we use horizontal flip and brightness jittering.

Inference. We use the first template and multiple online templates together with the current search region as input of MixFormer. The dynamic templates are updated when the update interval of 200 is reached by default. The template with the highest predicted score in the interval is selected to substitute the previous one.

4.2. Comparison with the state-of-the-art trackers

We verify the performance of our proposed MixFormer-1k, MixFormer-22k, and MixFormer-L on five benchmarks, including VOT2020 [26], LaSOT [17], TrackingNet [41], GOT10k [23], UAV123 [40].

VOT2020. VOT2020 [26] consists of 60 videos with several challenges including fast motion, occlusion, etc. As shown in Table 1, MixFormer-L achieves the top-ranked performance on EAO criteria of 0.555, which outperforms the transformer tracker STARK with a large margin of 5% of EAO. MixFormer-22k also outperforms other trackers including RPT (VOT2020 short-term challenge winner).

LaSOT. LaSOT [17] has 280 videos in its test set. We evaluate our MixFormer on the test set to validate its long-term capability. The Table 3 shows that our MixFormer surpasses all other trackers with a large margin. Specifically, MixFormer-L achieves the top-ranked performance on NP of 79.9%, surpassing STARK by 2.9% even without multi-layers feature aggregation.

TrackingNet. TrackingNet [41] provides over 30K videos with more than 14 million dense bounding box annotations. The videos are sampled from YouTube, covering target categories and scenes in real life. We validate MixFormer on its test set. From Table 3, we find that our MixFormer-22k and MixFormer-L set a new state-of-the-art performance on the large scale benchmark.

GOT10k. GOT10k [23] is a large-scale dataset with over 10000 video segments and has 180 segments for the test set. Apart from generic classes of moving objects and motion patterns, the object classes in the train and test set are zero-overlapped. As shown in Table 3, our MixFormer-GOT obtain state-of-the-art performance on the test split.

Table 3:

Method	LaSOT			TrackingNet			GOT-10k			UAV123	
	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$	AUC	P
MixFormer-L	**70.1**	**79.9**	**76.3**	**83.9**	**88.9**	**83.1**	<u>75.6</u>	<u>85.7</u>	<u>72.8</u>	69.5	**91.0**
MixFormer-22k	**69.2**	**78.7**	**74.7**	**83.1**	**88.1**	**81.6**	<u>72.6</u>	<u>82.2</u>	<u>68.8</u>	**70.4**	**91.8**
MixFormer-GOT	-	-	-	-	-	-	**70.7**	**80.0**	**67.8**	-	-
MixFormer-1k	67.9	77.3	73.9	82.6	87.7	81.2	**71.2**	**79.9**	**65.8**	68.7	89.5
STARK [57]	67.1	77.0	-	82.0	86.9	-	68.8	78.1	64.1	-	-
KeepTrack [39]	67.1	77.2	70.2	-	-	-	-	-	-	**69.7**	-
DTT [59]	60.1	-	-	79.6	85.0	78.9	63.4	74.9	51.4	-	-
SAOT [65]	61.6	70.8	-	-	-	-	64.0	75.9	-	-	-
AutoMatch [63]	58.2	-	59.9	76.0	-	72.6	65.2	76.6	54.3	-	-
TREG [10]	64.0	74.1	-	78.5	83.8	75.0	66.8	77.8	57.2	66.9	88.4
DualTFR [54]	63.5	72.0	66.5	80.1	84.9	-	-	-	-	68.2	-
TransT [7]	64.9	73.8	69.0	81.4	86.7	80.3	67.1	76.8	60.9	69.1	-
TrDiMP [49]	63.9	-	61.4	78.4	83.3	73.1	67.1	77.7	58.3	67.5	-
STMTracker [19]	60.6	69.3	63.3	80.3	85.1	76.7	64.2	73.7	57.5	64.7	-
SiamR-CNN [47]	64.8	72.2	-	81.2	85.4	80.0	64.9	72.8	59.7	64.9	83.4
PrDiMP [13]	59.8	68.8	60.8	75.8	81.6	70.4	63.4	73.8	54.3	68.0	-
OCEAN [64]	56.0	65.1	56.6	-	-	-	61.1	72.1	47.3	-	-
FCOT [9]	57.2	67.8	-	75.4	82.9	72.6	63.4	76.6	52.1	65.6	87.3
SiamGAT [20]	53.9	63.3	53.0	-	-	-	62.7	74.3	48.8	64.6	84.3
CGACD [16]	51.8	62.6	-	71.1	80.0	69.3	-	-	-	63.3	83.3
SiamFC++ [56]	54.4	62.3	54.7	75.4	80.0	70.5	59.5	69.5	47.9	-	-
MAML [48]	52.3	-	-	75.7	82.2	72.5	-	-	-	-	-
D3S [35]	-	-	-	72.8	76.8	66.4	59.7	67.6	46.2	-	-
DiMP [3]	56.9	65.0	56.7	74.0	80.1	68.7	61.1	71.7	49.2	65.4	-
ATOM [12]	51.5	57.6	50.5	70.3	77.1	64.8	55.6	63.4	40.2	64.3	-
SiamRPN++ [28]	49.6	56.9	49.1	73.3	80.0	69.4	51.7	61.6	32.5	61.0	80.3
MDNet [42]	39.7	46.0	37.3	60.6	70.5	56.5	29.9	30.3	9.9	52.8	-
SiamFC [2]	33.6	42.0	33.9	57.1	66.3	53.3	34.8	35.3	9.8	48.5	69.3

Table 3. State-of-the-art comparison on TrackingNet [41], LaSOT [17], GOT-10k [23] and UAV123 [40]. The best two results are shown in **red** and **blue** fonts. The <u>underline</u> results of GOT-10k are not considered in the comparison, since the models are trained with datasets other than GOT-10k. MixFormer-1k is the model pretrained with ImageNet-1k. Others are pretrained with ImageNet-22k.

UAV123. UAV123 [40] is a large dataset containing 123 Sequences with average sequence length of 915 frames, which is captured from low-altitude UAVs. Table 3 shows our results on UAV123 dataset. Our MixFormer-22k and MixFormer-L outperforms all other trackers.

4.3. Exploration Studies

To verify the effectiveness and give a thorough analysis on our proposed MixFormer, we perform a detailed ablation study on the large-scale LaSOT dataset.

Simultaneous process vs. Separate process. As the core part of our MixFormer is to unify the procedure of feature extraction and target information integration, we compare it to the separate processing architecture (e.g. TransT [6]). The comparison results are shown in Table 4 #1, #2, #3 and #8. Experiments of #1 and #2 are end-to-end trackers comprising a self-attention based backbone, n cross attention modules to perform information integration and a corner head. #3 is the tracker with CVT as backbone and TransT's ECA+CFA(4) as interaction. Experiment of #8 is our Mix-Former **without** multiple online templates and asymmetric mechanism, denoted by *MixFormer-Base*. MixFormer-Base largely increases the model of #1 (using one CAM) and #2

#	Backbone	Integration	Head	Params.	FLOPs	AUC
1	SAM(21)	CAM(1)	Corner	37.35M	20.69G	59.8
2	SAM(21)	CAM(3)	Corner	40.92M	22.20G	60.5
3	SAM(21)	ECA+CFA(4)	Corner	49.75M	27.81G	66.9
4	SAM(20)+MAM(1)	-	Corner	35.57M	19.97G	65.8
5	SAM(15)+MAM(6)	-	Corner	35.67M	20.02G	66.2
6	SAM(10)+MAM(11)	-	Corner	35.77M	20.07G	67.4
7	SAM(5)+MAM(16)	-	Corner	35.87M	20.12G	68.1
8	MAM(21)	-	Corner	35.97M	20.85G	**68.4**
9	MAM(21)	-	Query	31.46M	19.13G	66.0

Table 4. Analysis of the MAM based framework. '-' denotes the component is not used. SAM represents for self attention module, CAM for cross attention module and MAM for the proposed mixed attention module. The numbers in brackets represent the number of the blocks. Performance is evaluated on LaSOT.

(using three CAMs) by 8.6% and 7.9% with smaller parameters and FLOPs. This demonstrates the effectiveness of unified feature extraction and information integration, as both of them would benefit each other.

Study on stages of MAM. To further verify the effectiveness of the MAMs, we conduct experiments as in Table 4 #4, #5, #6, #7 and #8, to investigate the performance of different numbers of MAM in our MixFormer. We compare our MAM with the self-attention operations (SAM) with out cross-branch information communication. We find that more MAMs contribute to higher AUC score. It indi-

	Asymmetric.	FPS (1080Ti)	AUC
MixFormer-Base	-	19	**68.4**
MixFormer-Base	✓	25	68.1

Table 5. Ablation for asymmetric mixed attention mechanism.

	Online	Score	AUC
MixFormer	-	-	68.1
MixFormer	✓	-	66.6
MixFormer	✓	✓	**69.2**

Table 6. Ablation for online templates update mechanism.

	Pretrain	Train	AUC
MixFormer	ImageNet-1k	Whole	67.9
MixFormer	ImageNet-22k	Whole	**69.2**
MixFormer	ImageNet-22k	GOT-10k	62.1

Table 7. Study on pretraining and training datasets. 'Whole' denotes for using the whole datasets including GOT-10k, LaSOT, TrackingNet and COCO.

cates that extensive target-aware feature extraction and hierarchical information integration play a critical role to construct an effective tracker, which is realized by the iterative MAM. Especially, when the number of MAM reaches 16, the performance reaches 68.1, which is comparable to the MixFormer-Base containing 21 MAMs.

Study on localization head. To verify the generalization ability of our MAM backbone, we evaluate the MixFormer-Base with two types of localization head as described in Section 3.2 (fully convolutional head vs. query based head). The results are shown as in Table 4 #8 and #9 for the corner head and the query-base head respectively. MixFormer-Base with the fully convolutional corner head outperforms that of the query-based head. Especially, MixFormer-Base with corner head surpass all the other state-of-the-art trackers even without any post-processing and online templates. Besides, MixFormer-Base with the query head, a pure transformer-based tracking framework, obtains a comparable AUC of 66.0 with STARK-ST and KeepTrack [39] and far exceed query-head STARK-ST of 63.7. It demonstrates the generalization ability of our MAM backbone.

Study on asymmetric MAM. The asymmetric MAM is used to reduce computational cost and allows for usage of multiple templates during online tracking. As shown in Table 5, the asymmetric MixFormer-Base increases the running speed of 24% while achieving a comparable performance, which demonstrates asymmetric MAM is important for building an efficient tracker.

Study on online template update. As demonstrated in Table 6, MixFormer with online templates, sampled by a fixed update interval, performs worse than that with only the first template, and the online MixFormer with our score prediction module achieves the best AUC score. It suggests that selecting reliable templates with our score prediction module is of vital importance.
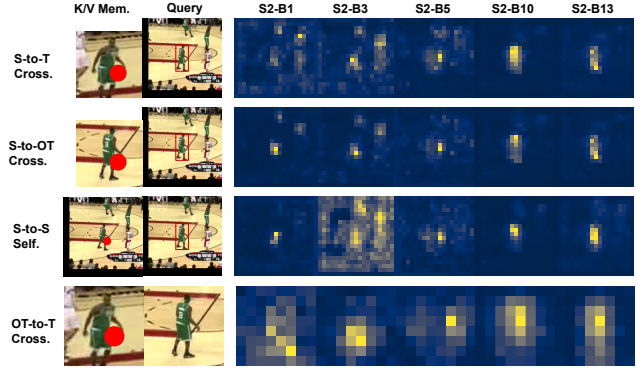

Figure 5. Visualization results of different attention weights on *Basketball* of OTB100. **S-to-t** is search-to-template cross attention, **S-to-OT** is search-to-online-template cross attention, **S-to-S** is self attention of search region and **OT-to-T** is online-template-to-template cross attention. **S**i-**B**j represents for Stage-i and Block-j of MixFormer. Best viewed with zooming in.

Study on training and pre-training datasets. To verify the generalization ablility of our MixFormer, we conduct an analysis on different pre-training and training datasets, as shown in Table 7. MixFormer pretrained by ImageNet-1k still outperforms all the SOTA trackers (e.g., TransT [6], KeepTrack [39], STARK [57]), even without post-processing and multi-layer feature aggregation. In addition, MixFormer trained with GOT-10k also achieves an impressive AUC of 62.1, which outperforms a majority of trackers trained with the whole tracking datasets.

Visualization of attention maps. To explore how the mixed attention works in MixFormer backbone, we visualize some attention maps in Fig. 5. From the four types of attention maps, we derive that: (i) distractors in background get suppressed layer by layer, (ii) online templates may be more adaptive to appearance variation and help to discriminate the target, (iii) the foreground of multiple templates can be augmented by mutual cross attention, (iv) a certain position tends to interact with the surrounding local patch.

5. Conclusion

We have presented MixFormer, an end-to-end tracking framework with iterative mixed attention, aiming to unify the feature extraction and target integration and result in a neat and compact tracking pipeline. Mixed attention module performs both feature extraction and mutual interaction for target template and search area. In empirical evaluation, MixFormer shows a notable improvement over other prevailing trackers for short-term tracking. In the future, we consider extending MixFormer to multiple object tracking.

Figure 6. Visualization results of different attention weights on *car-2* of LaSOT. **S-to-t** is search-to-template cross attention, **S-to-OT** is search-to-online-template cross attention, **S-to-S** is self attention of search region and **OT-to-T** is online-template-to-template cross attention. **S**i-**B**j represents for Stage-i and Block-j of MixFormer. Best viewed with zooming in.

Appendix

In this appendix, we first provide more results and analysis on OTB100 [53] and LaSOT [17] datasets. Then we give more visualization results of the attention weights on LaSOT. Finally, we provide more training details.

A. More Results

OTB-100. OTB100 [53] is a commonly used benchmark, which evaluates performance on Precision and AUC scores. Figure. 7 presents results of our trackers on both two metrics on OTB-100 benchmark. MixFormer-L reaches competitive performance w.r.t. state-of-the-art trackers, surpassing the transformer tracker TransT by 1.3% on AUC score. Besides, MixFormer-L is slightly higher than MixFormer.

LaSOT. LaSOT [17] has 280 videos in its test set. We evaluate our MixFormer on the test set to validate its long-term capability. To give a further analysis, we provide Success plot and Precision plot for LaSOT in Fig. 8. It proves that improvement is due to both higher accuracy and robustness.

Figure 7. State-of-the-art comparison on the OTB100 dataset. Best viewed with zooming in.

Figure 8. State-of-the-art comparison on the LaSOT dataset.

B. More Visualization Results

In this section, we provide more visualization results of attention weights on *car-2* of LaSOT test dataset in Fig. 6. From the example, we can arrive at the same conclusion with section 4.3. Besides, from the last two lines, we infer that the features of last two blocks tend to adapt to the bounding box prediction head.

C. Training Details

We propose a 320x320 search region plus two 128x128 input images to make a fair comparison with prevailing trackers (e.g., Siamese-based trackers, STARK and TransT). Generally, we use 8 Tesla V100 GPUSs to train MixFormer with batch size of 32. MixFormer can also be trained on 8 2080Ti GPUs having only 11GB memory, with batch size of 8 per GPU. We use CvT21 and CvT24-W as the pretrained model for MixFormer and MixFormer-L respectively. We apply gradient clip strategy with the clip normalization rate of 0.1. For training stage-1 of MixFormer (i.e., MixFormer without SPM), we use GIoU loss and L_1 loss, with the weights of 2.0 and 5.0 respectively. Besides, the Batch Normalization layers of MixFormer backbone are frozen during the whole training process. For SPM training process, the backbone and corner-based localization head are frozen and the batch size is 32. SPM is trained for 40 epochs with the learning rate decays at 30 epochs.

References

[1] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[2] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, 2016. 1, 2, 7

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 1, 2, 6, 7

[4] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018. 1

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020. 3, 5

[6] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 4, 7, 8

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021. 2, 3, 5, 7

[8] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, June 2020. 1

[9] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Fully convolutional online tracking. *CoRR*, 2020. 1, 2, 7

[10] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Target transformed regression for accurate tracking. *CoRR*, 2021. 1, 2, 3, 7

[11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, 2017. 2

[12] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: accurate tracking by overlap maximization. In *CVPR*, 2019. 1, 2, 7

[13] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020. 7

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009. 6

[15] 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. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021*. 2

[16] Fei Du, Peng Liu, Wei Zhao, and Xianglong Tang. Correlation-guided attention for corner detection based visual tracking. In *CVPR*, 2020. 3, 7

[17] 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*, 2019. 2, 6, 7, 9

[18] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019. 1

[19] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 7

[20] Dongyan Guo, Yanyan Shao, Ying Cui, Zhenhua Wang, Liyan Zhang, and Chunhua Shen. Graph attention tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021. 7

[21] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*, June 2020. 2

[22] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015. 1, 2, 6

[23] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5):1562–1577, 2021. 2, 6, 7

[24] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 6

[26] Matej Kristan, Ales Leonardis, and et. al. The eighth visual object tracking VOT2020 challenge results. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision - ECCV 2020 Workshops*, 2020. 2, 6

[27] Matej Kristan, Jiří Matas, Aleš Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Cehovin, Alan Lukežič, Ondrej Drbohlav, Jani Käpylä, Gustav Häger, Song Yan, Jinyu Yang, Zhongqun Zhang, and Gustavo Fernández. The ninth visual object tracking vot2021 challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2021. 3

[28] 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*, 2019. 2, 5, 7

[29] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 1, 2

[30] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[31] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss V2: learning reliable localization quality estimation for dense object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 2021. 5

[32] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *CoRR*, abs/2104.05707, 2021. 2

[33] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 6

[34] Liwei Liu, Junliang Xing, Haizhou Ai, and Xiang Ruan. Hand posture recognition using finger geometric feature. In *ICPR*, 2012. 1

[35] Alan Lukezic, Jiri Matas, and Matej Kristan. D3S - A discriminative single shot segmentation tracker. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 7131–7140. Computer Vision Foundation / IEEE, 2020. 6, 7

[36] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *CVPR*, 2017. 1, 2

[37] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017. 1

[38] Ziang Ma, Linyuan Wang, Haitao Zhang, Wei Lu, and Jun Yin. RPT: learning point set representation for siamese visual tracking. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision - ECCV 2020 Workshops*, Lecture Notes in Computer Science, 2020. 6

[39] 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*, 2021. 7, 8

[40] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, 2016. 2, 6, 7

[41] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018. 2, 6, 7

[42] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 1, 7

[43] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV*. IEEE, 2019. 6

[44] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 2019. 5

[45] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson W. H. Lau, and Ming-Hsuan Yang. VITAL: visual tracking via adversarial learning. In *CVPR*, 2018. 1

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *NIPS*, 2017. 1, 3

[47] Paul Voigtlaender, Jonathon Luiten, Philip H.S. Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 7

[48] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020. 7

[49] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 7

[50] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H. S. Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 6

[51] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[52] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *CoRR*, abs/2103.15808, 2021. 2, 5, 6

[53] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1834–1848, 2015. 9

[54] Fei Xie, Chunyu Wang, Guangting Wang, Wankou Yang, and Wenjun Zeng. Learning tracking representations via dual-branch fully transformer networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2021. 6, 7

[55] Junliang Xing, Haizhou Ai, and Shihong Lao. Multiple human tracking based on multi-view upper-body detection and discriminative learning. In *ICPR*, 2010. 1

[56] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 1, 7

[57] 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 (ICCV), 2021. 1, 2, 3, 4, 5, 6, 7, 8

[58] Bin Yan, Xinyu Zhang, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Alpha-refine: Boosting tracking performance by precise bounding box estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 2021. 6

[59] Bin Yu, Ming Tang, Linyu Zheng, Guibo Zhu, Jinqiao Wang, Hao Feng, Xuetao Feng, and Hanqing Lu. High-performance discriminative tracking with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 7

[60] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R. Scott. Deformable siamese attention networks for visual object tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6727–6736. Computer Vision Foundation / IEEE, 2020. 3

[61] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[62] Zhipeng Zhang, Yufan Liu, Bing Li, Weiming Hu, and Houwen Peng. Toward accurate pixelwise object tracking via attention retrieval. *IEEE Transactions on Image Processing*, 30:8553–8566, 2021. 6

[63] Zhipeng Zhang, Yihao Liu, Xiao Wang, Bing Li, and Weiming Hu. Learn to match: Automatic matching network design for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 7

[64] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 1, 7

[65] Zikun Zhou, Wenjie Pei, Xin Li, Hongpeng Wang, Feng Zheng, and Zhenyu He. Saliency-associated object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 7

[66] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018. 2