AiATrack: Attention in Attention for Transformer Visual Tracking

Shenyuan Gao[1], Chunluan Zhou[2], Chao Ma[3], Xinggang Wang[1], Junsong Yuan[4]

[1] Huazhong University of Science and Technology [2] Wormpex AI Research
[3] Shanghai Jiao Tong University [4] State University of New York at Buffalo
shenyuangao@gmail.com, czhou002@e.ntu.edu.sg
chaoma@sjtu.edu.cn, xgwang@hust.edu.cn, jsyuan@buffalo.edu

Abstract. Transformer trackers have achieved impressive advancements recently, where the attention mechanism plays an important role. However, the independent correlation computation in the attention mechanism could result in noisy and ambiguous attention weights, which inhibits further performance improvement. To address this issue, we propose an attention in attention (AiA) module, which enhances appropriate correlations and suppresses erroneous ones by seeking consensus among all correlation vectors. Our AiA module can be readily applied to both self-attention blocks and cross-attention blocks to facilitate feature aggregation and information propagation for visual tracking. Moreover, we propose a streamlined Transformer tracking framework, dubbed AiATrack, by introducing efficient feature reuse and target-background embeddings to make full use of temporal references. Experiments show that our tracker achieves state-of-the-art performance on six tracking benchmarks while running at a real-time speed. Code and models are publicly available at https://github.com/Little-Podi/AiATrack.

Keywords: Visual Tracking, Attention Mechanism, Vision Transformer

1 Introduction

Visual tracking is one of the fundamental tasks in computer vision. It has gained increasing attention because of its wide range of applications [35,18]. Given a target with bounding box annotation in the initial frame of a video, the objective of visual tracking is to localize the target in successive frames. Over the past few years, Siamese trackers [2,32,31,58], which regards the visual tracking task as a one-shot matching problem, have gained enormous popularity. Recently, several trackers [48,8,54,6,52,51] have explored the application of the Transformer [47] architecture and achieved promising performance.

The crucial components in a typical Transformer tracking framework [48,8,54] are the attention blocks. As shown in Fig. 1, the feature representations of the reference frame and search frame are enhanced via self-attention blocks, and the correlations between them are bridged via cross-attention blocks for target prediction in the search frame. The Transformer attention [47] takes queries and

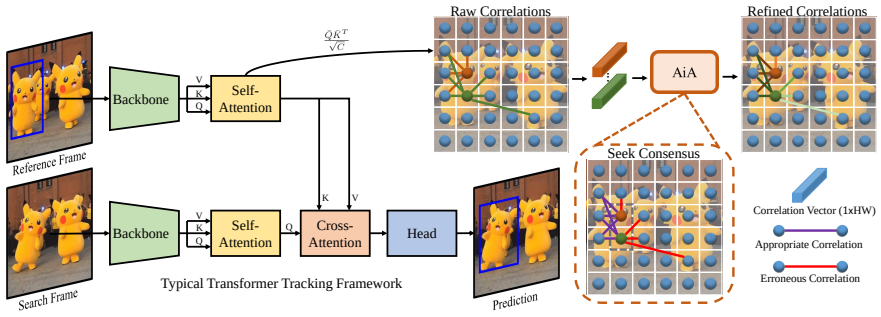

Fig. 1. Motivation of the proposed method. The left part of the figure shows a typical Transformer tracking framework. On the right, the nodes denote features at different positions in a feature map. These nodes serve as queries and keys for a self-attention block. The links between nodes represent the correlations between queries and keys in the attention mechanism. Some correlations of the green node is erroneous since it is linked to the nodes at irrelevant positions. By applying the proposed module to the raw correlations, we can seek consensus from the correlations of other nodes (*e.g.* the brown node) that can provide supporting cues for the appropriate correlations. By this means, the quality of the correlations can be refined.

a set of key-value pairs as input and outputs linear combinations of values with weights determined by the correlations between queries and the corresponding keys. The correlation map is computed by the scaled dot products between queries and keys. However, the correlation of each query-key pair is computed independently, which ignores the correlations of other query-key pairs. This could introduce erroneous correlations due to imperfect feature representations or the existence of distracting image patches in a background clutter scene, resulting in noisy and ambiguous attention weights as visualized in Fig. 4.

To address the aforementioned issue, we propose a novel attention in attention (AiA) module, which extends the conventional attention [47] by inserting an inner attention module. The introduced inner attention module is designed to refine the correlations by seeking consensus among all correlation vectors. The motivation of the AiA module is illustrated in Fig. 1. Usually, if a key has a high correlation with a query, some of its neighboring keys will also have relatively high correlations with that query. Otherwise, the correlation might be noise. Motivated by this, we introduce the inner attention module to utilize these informative cues. Specifically, the inner attention module takes the raw correlations as queries, keys, and values and adjusts them to enhance the appropriate correlations of relevant query-key pairs and suppress the erroneous correlations of irrelevant query-key pairs. We show that the proposed AiA module can be readily inserted into the self-attention blocks to enhance feature aggregation and into the cross-attention block to facilitate information propagation, both of which are very important in a Transformer tracking framework. As a result, the overall tracking performance can be improved.

How to introduce the long-term and short-term references is still an open problem for visual tracking. With the proposed AiA module, we present AiATrack, a streamlined Transformer framework for visual tracking. Unlike previous practices [56,19,48,52], which need an extra computational cost to process the selected reference frame during the model update, we directly reuse the cached features which are encoded before. An IoU prediction head is introduced for selecting high-quality short-term references. Moreover, we introduce learnable target-background embeddings to distinguish the target from the background while preserving the contextual information. With these designs, the proposed AiATrack can efficiently update short-term references and effectively exploit the long-term and short-term references for visual tracking.

We verify the effectiveness of our method by conducting comprehensive experiments on six prevailing benchmarks covering various kinds of tracking scenarios. Without bells and whistles, the proposed AiATrack sets new state-of-the-art results on these benchmarks with a real-time speed of 38 frames per second (fps).

In summary, the main contributions of our work are three-fold:

- We propose a novel attention in attention (AiA) module, which can mitigate noise and ambiguity in the conventional attention mechanism [47] and improve tracking performance by a notable margin.
- We present a neat Transformer tracking framework with the reuse of encoded features and the introduction of target-background embeddings to efficiently and effectively leverage temporal references.
- We perform extensive experiments and analyses to validate the effectiveness of our designs. The proposed AiATrack achieves state-of-the-art performance on six widely used benchmarks.

2 Related Work

2.1 Visual Tracking

Recently, Transformer [47] has shown impressive performance in computer vision [7,59,14]. It aggregates information from sequential inputs to capture global context by an attention mechanism. Some efforts [55,21,19] have been made to introduce the attention structure to visual tracking. Recently, several works [48,8,54,6,52,51] apply Transformer architecture to visual tracking. Despite their impressive performance, the potential of Transformer trackers is still limited by the conventional attention mechanism. To this end, we propose a novel attention module, namely, attention in attention (AiA), to further unveil the power of Transformer trackers.

How to adapt the model to the appearance change during tracking has also been investigated by previous works [11,3,56,10,4,19,48,52]. A straightforward solution is to update the reference features by generation [56] or ensemble [19,48,52]. However, most of these methods need to resize the reference frame and re-encode the reference features, which may sacrifice computational efficiency. Following discriminative correlation filter (DCF) method [24], another

family of approaches [11,3] optimize the network parameters during the inference. However, they need sophisticated optimization strategies with a sparse update to meet real-time requirements. In contrast, we present a new framework that can efficiently reuse the encoded features. Moreover, a target-background embedding assignment mechanism is also introduced. Different from [20,53,30], our target-background embeddings are directly introduced to distinguish the target and background regions and provide rich contextual cues.

2.2 Attention Mechanism

Represented by non-local operation [49] and Transformer attention [47], attention mechanism has rapidly received great popularity over the past few years. Recently, Transformer attention has been introduced to computer vision as a competitive architecture [7,59,14]. In vision tasks, it usually acts as a dynamic information aggregator in spatial and temporal domains. There are some works [26,27] that focus on solving existing issues in the conventional attention mechanism. Unlike these, in this paper, we try to address the noise and ambiguity issue in conventional attention mechanism by seeking consensus among correlations with a global receptive field.

2.3 Correlation as Feature

Treating correlations as features has been explored by several previous works [45,44,5,43,33,38,42,9,4]. In this paper, we use correlations to refer to the matching results of the pixels or regions. They can be obtained by squared difference, cosine similarity, inner product, *etc*. Several efforts have been made to recalibrate the raw correlations by processing them as features through hand-crafted algorithms [44,5] or learnable blocks [43,38,33,4,42,9]. To our best knowledge, we introduce this insight to the attention mechanism for the first time, making it a unified block for feature aggregation and information propagation in Transformer visual tracking.

3 Method

3.1 Attention in Attention

To present our attention in attention module, we first briefly revisit the conventional attention block in vision [14,7]. As illustrated in Fig. 2(a), it takes a query and a set of key-value pairs as input and produces an output which is a weighted sum of the values. The weights assigned to the values are computed by taking the softmax of the scaled dot products between the query and the corresponding keys. Denote queries, keys and values by $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V} \in \mathbb{R}^{HW \times C}$ respectively. The conventional attention can be formulated as

$$\text{ConvenAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\text{Softmax}\left(\frac{\bar{\mathbf{Q}}\bar{\mathbf{K}}^{\mathbf{T}}}{\sqrt{C}}\right)\bar{\mathbf{V}})\mathbf{W_o} \tag{1}$$

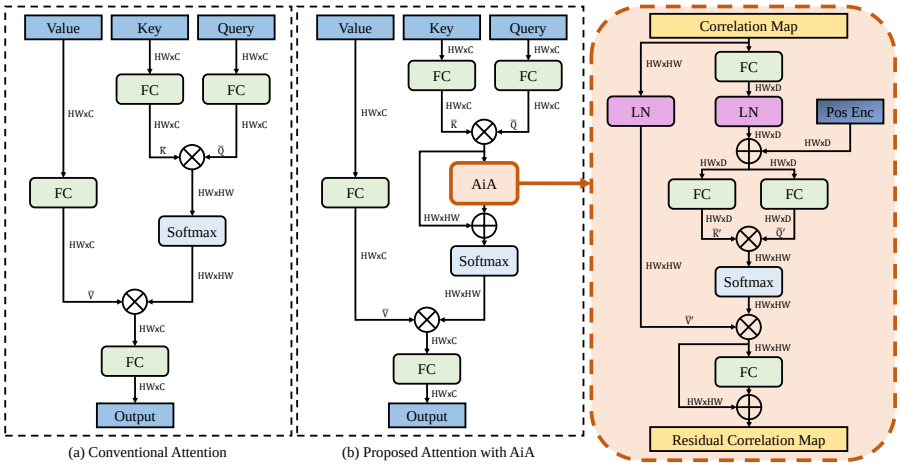

Fig. 2. Structures of conventional attention and the proposed attention in attention (AiA) module. $\otimes$ denotes matrix multiplication and $\oplus$ denotes element-wise addition. The numbers beside arrows are feature dimensions which do not include the batch size. Matrix transpose operations are omitted for brevity.

where $\bar{\mathbf{Q}} = \mathbf{Q}\mathbf{W_q}$, $\bar{\mathbf{K}} = \mathbf{K}\mathbf{W_k}$, $\bar{\mathbf{V}} = \mathbf{V}\mathbf{W_v}$ are different linear transformations. Here, $\mathbf{W_q}$, $\mathbf{W_k}$, $\mathbf{W_v}$ and $\mathbf{W_o}$ denote the linear transform weights for queries, keys, values, and outputs, respectively.

However, in the conventional attention block, the correlation of each query-key pair in the correlation map $\mathbf{M} = \frac{\bar{\mathbf{Q}}\bar{\mathbf{K}}^{\mathbf{T}}}{\sqrt{C}} \in \mathbb{R}^{HW \times HW}$ is computed independently, which ignores the correlations of other query-key pairs. This correlation computation procedure may introduce erroneous correlations due to imperfect feature representations or the existence of distracting image patches in a background clutter scene. These erroneous correlations could result in noisy and ambiguous attentions as visualized in Fig. 4. They may unfavorably affect the feature aggregation in self-attention and the information propagation in cross-attention, leading to sub-optimal performance for a Transformer tracker.

To address the aforementioned problem, we propose a novel attention in attention (AiA) module to improve the quality of the correlation map $\mathbf{M}$. Usually, if a key has a high correlation with a query, some of its neighboring keys will also have relatively high correlations with that query. Otherwise, the correlation might be a noise. Motivated by this, we introduce the AiA module to utilize the informative cues among the correlations in $\mathbf{M}$. The proposed AiA module seeks the correlation consistency around each key to enhance the appropriate correlations of relevant query-key pairs and suppress the erroneous correlations of irrelevant query-key pairs.

Specifically, we introduce another attention module to refine the correlation map $\mathbf{M}$ before the softmax operation as illustrated in Fig. 2(b). As the newly introduced attention module is inserted into the conventional attention block,

we call it an inner attention module, forming an attention in attention structure. The inner attention module itself is a variant of the conventional attention. We consider columns in $\mathbf{M}$ as a sequence of correlation vectors which are taken as queries $\mathbf{Q}'$, keys $\mathbf{K}'$ and values $\mathbf{V}'$ by the inner attention module to output a residual correlation map.

Given the input $\mathbf{Q}'$, $\mathbf{K}'$ and $\mathbf{V}'$, we first generate transformed queries $\bar{\mathbf{Q}}'$ and keys $\bar{\mathbf{K}}'$ as illustrated in the right block of Fig. 2(b). To be specific, a linear transformation is first applied to reduce the dimensions of $\mathbf{Q}'$ and $\mathbf{K}'$ to $HW \times D$ ($D \ll HW$) for computational efficiency. After normalization [1], we add 2-dimensional sinusoidal encoding [14,7] to provide positional cues. Then, $\bar{\mathbf{Q}}'$ and $\bar{\mathbf{K}}'$ are generated by two different linear transformations. We also normalize $\mathbf{V}'$ to generate the normalized correlation vectors $\bar{\mathbf{V}}'$, *i.e.* $\bar{\mathbf{V}}' = \text{LayerNorm}(\mathbf{V}')$. With $\bar{\mathbf{Q}}'$, $\bar{\mathbf{K}}'$ and $\bar{\mathbf{V}}'$, the inner attention module generates a residual correlation map by

$$\text{InnerAttn}(\mathbf{M}) = (\text{Softmax}\left(\frac{\bar{\mathbf{Q}}'\bar{\mathbf{K}}'^{\mathbf{T}}}{\sqrt{D}}\right)\bar{\mathbf{V}}')(1 + \mathbf{W}'_{\mathbf{o}}) \tag{2}$$

where $\mathbf{W}'_{\mathbf{o}}$ denotes linear transform weights for adjusting the aggregated correlations together with an identical connection.

Essentially, for each correlation vector in the correlation map $\mathbf{M}$, the AiA module generates its residual correlation vector by aggregating the raw correlation vectors. It can be seen as seeking consensus among the correlations with a global receptive field. With the residual correlation map, our attention block with AiA module can be formulated as

$$\text{AttninAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\text{Softmax}(\mathbf{M} + \text{InnerAttn}(\mathbf{M}))\bar{\mathbf{V}})\mathbf{W}_{\mathbf{o}} \tag{3}$$

For a multi-head attention block, we share the parameters of the AiA module between the parallel attention heads. It is worth noting that our AiA module can be readily inserted into both self-attenion and cross-attention blocks in a Transformer tracking framework.

3.2 Proposed Framework

With the proposed AiA module, we design a simple yet effective Transformer framework for visual tracking, dubbed AiATrack. Our tracker is comprised of a network backbone, a Transformer architecture, and two prediction heads as illustrated in Fig. 3. Given the search frame, the initial frame is taken as a long-term reference and an ensemble of several intermediate frames are taken as short-term references. The features of the long-term and short-term references and the search frame are extracted by the network backbone and then reinforced by the Transformer encoder. We also introduce learnable target-background embeddings to distinguish the target from background regions. The Transformer decoder propagates the reference features as well as the target-background embedding maps to the search frame. The output of the Transformer is then fed to a target prediction head and an IoU prediction head for target localization and short-term reference update, respectively.

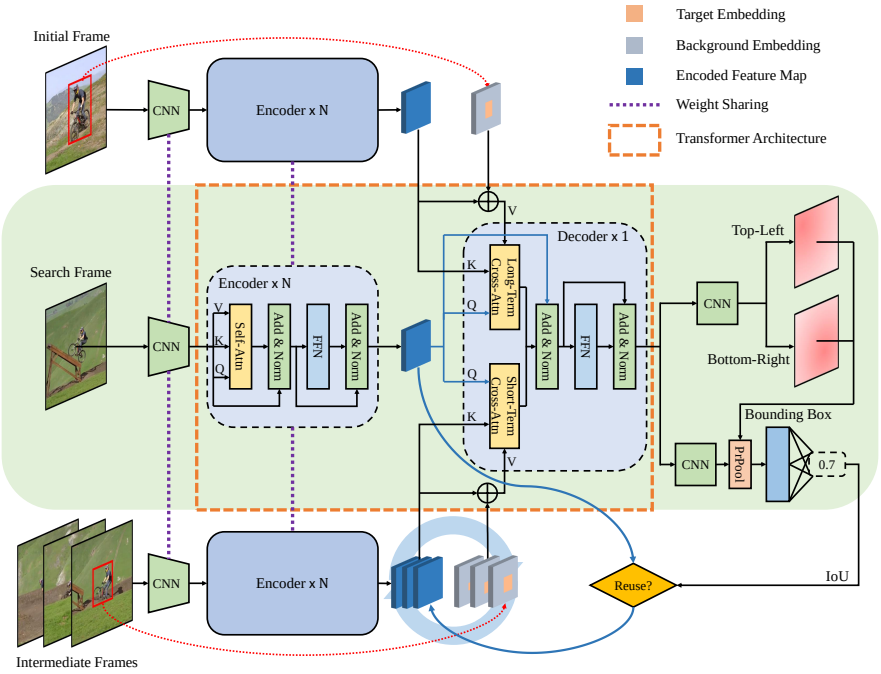

Fig. 3. Overview of the proposed Transformer tracking framework. The self-attention and cross-attention blocks are all equipped with the proposed AiA module. Note that only the components on the light green background need to be computed during the inference phase as described in Sec. 3.3.

Transformer Architecture. The Transformer encoder is adopted to reinforce the features extracted by the convolutional backbone. For the search frame, we flatten its features to obtain a sequence of feature vectors and add sinusoidal positional encoding as in [7]. The sequence of feature vectors is then taken by the Transformer encoder as its input. The Transformer encoder consists of several layer stacks, each of which is made up of a multi-head self-attention block and a feed-forward network. The self-attention block serves to capture the dependencies among all feature vectors to enhance the original features, and is equipped with the proposed AiA module. Similarly, this procedure is applied independently to the features of the reference frames using the same encoder.

The Transformer decoder propagates the reference information from the long-term and short-term references to the search frame. Different from the classical Transformer decoder [47], we remove the self-attention block for simplicity and introduce a two-branch cross-attention design as shown in Fig. 3 to retrieve the target-background information from long-term and short-term references. The long-term branch is responsible for retrieving reference information from the initial frame. Since the initial frame has the most reliable annotation of the tracking target, it is crucial for robust visual tracking. However, as the appearance of the

target and the background change through the video, the reference information from the long-term branch may not be up-to-date. This could cause tracker drift in some scenes. To address this problem, we introduce the short-term branch to utilize the information from the frames that are closer to the current frame. The cross-attention blocks of the two branches have the identical structure following the query-key-value design in the vanilla transformer [47]. We take the features of the search frame as queries and the features of the reference frames as keys. The values are generated by combining the reference features with target-background embedding maps, which will be described below. We also insert our AiA module into cross-attention for better reference information propagation.

Target-Background Embeddings. To indicate the target and background regions while preserving the contextual information, we introduce a target embedding $\mathcal{E}^{tgt} \in \mathbb{R}^C$ and a background embedding $\mathcal{E}^{bg} \in \mathbb{R}^C$, both of which are learnable. With $\mathcal{E}^{tgt}$ and $\mathcal{E}^{bg}$, we generate target-background embedding maps $\mathcal{E} \in \mathbb{R}^{HW \times C}$ for the reference frames with a negligible computational cost. Let's consider a location p in a $H \times W$ grid, the embedding assignment is formulated as

$$\mathcal{E}(p) = \begin{cases} \mathcal{E}^{tgt} & \text{if } p \text{ falls in the target region} \\ \mathcal{E}^{bg} & \text{otherwise} \end{cases} \tag{4}$$

Afterward, we attach the target-background embedding maps to the reference features and feed them to cross-attention blocks as values. The target-background embedding maps enrich the reused appearance features by providing contextual cues.

Prediction Heads. As described above, our tracker has two prediction heads. The target prediction head is adopted from [52]. Specifically, the decoded features are fed into a two-branch fully-convolutional network which outputs two probability maps for the top-left and the bottom-right corners of the target bounding box. The predicted box coordinates are then obtained by computing the expectations of the probability distributions of the two corners.

To adapt the model to the appearance change during tracking, the tracker needs to keep the short-term references up-to-date by selecting reliable references which contain the target. Moreover, considering our embedding assignment mechanism in Eq. 4, the bounding box of the selected reference frame should be as accurate as possible. Inspired by IoU-Net [28] and ATOM [11], for each predicted bounding box, we estimate its IoU with the ground truth via an IoU prediction head. The features inside the predicted bounding box are passed to a Precise RoI Pooling layer whose output is taken by a fully connected network to produce an IoU prediction. The predicted IoU is then used to determine whether to include the search frame as a new short-term reference.

We train the two prediction heads jointly. The loss of target prediction is defined by the combination of GIoU loss [41] and L1 loss between the predicted bounding box and the ground truth. The training examples of the IoU prediction head are generated by sampling bounding boxes around the ground truths. The loss of IoU prediction is defined by mean squared error. We refer readers to the supplementary material for more details about training.

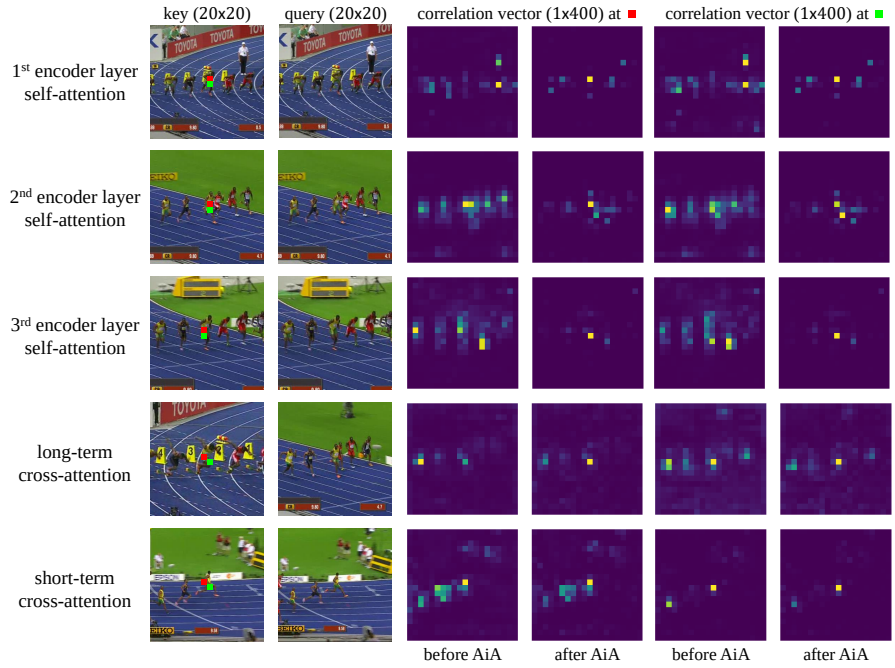

Fig. 4. Visualization of the effect of the proposed AiA module. We visualize several representative correlation vectors before and after the refinement by the AiA module. The visualized correlation vectors are reshaped according to the spatial positions of queries. We select the correlation vectors of keys corresponding to the target object regions in the first column. It can be observed that the erroneous correlations are effectively suppressed and the appropriate ones are enhanced with the AiA module.

3.3 Tracking with AiATrack

Given the initial frame with ground truth annotation, we initialize the tracker by cropping the initial frame as long-term and short-term references and pre-computing their features and target-background embedding maps. For each subsequent frame, we estimate the IoU score of the bounding box predicted by target prediction head for model update. The update procedure is more efficient than the previous practices [19,48,52], as we directly reuse the encoded features. Specifically, if the estimated IoU score of the predicted bounding box is higher than the pre-defined threshold, we generate the target-background embedding map for the current search frame and store the embedding map in a memory cache together with its encoded features. For each new-coming frame, we uniformly sample several short-term reference frames and concatenate their features and embedding maps from the memory cache to update the short-term reference ensemble. The latest reference frame in the memory cache is always sampled as it is closest to the current search frame. The oldest reference frame in the memory cache will be popped out if the maximum cache size is reached.

Tracker	Source	LaSOT [17]			TrackingNet [40]			GOT-10k [25]		
		AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$SR_{0.75}$	$SR_{0.5}$
AiATrack	Ours	69.0	79.4	73.8	82.7	87.8	80.4	69.6	63.2	80.0
STARK-ST50 [52]	ICCV2021	66.4	76.3	71.2	81.3	86.1	78.1	68.0	62.3	77.7
KeepTrack [36]	ICCV2021	67.1	77.2	70.2	-	-	-	-	-	-
DTT [54]	ICCV2021	60.1	-	-	79.6	85.0	78.9	63.4	51.4	74.9
TransT [8]	CVPR2021	64.9	73.8	69.0	81.4	86.7	80.3	67.1	60.9	76.8
TrDiMP [48]	CVPR2021	63.9	-	61.4	78.4	83.3	73.1	67.1	58.3	77.7
TrSiam [48]	CVPR2021	62.4	-	60.0	78.1	82.9	72.7	66.0	57.1	76.6
KYS [4]	ECCV2020	55.4	63.3	-	74.0	80.0	68.8	63.6	51.5	75.1
Ocean-online [58]	ECCV2020	56.0	65.1	56.6	-	-	-	61.1	47.3	72.1
Ocean-offline [58]	ECCV2020	52.6	-	52.6	-	-	-	59.2	-	69.5
PrDiMP50 [12]	CVPR2020	59.8	68.8	60.8	75.8	81.6	70.4	63.4	54.3	73.8
SiamAttn [55]	CVPR2020	56.0	64.8	-	75.2	81.7	-	-	-	-
DiMP50 [3]	ICCV2019	56.9	65.0	56.7	74.0	80.1	68.7	61.1	49.2	71.7
SiamRPN++ [31]	CVPR2019	49.6	56.9	49.1	73.3	80.0	69.4	51.7	32.5	61.6

Table 1. State-of-the-art comparison on LaSOT, TrackingNet, and GOT-10k. The best two results are shown in red and blue, respectively. All the trackers listed above adopt ResNet-50 pre-trained on ImageNet-1k as network backbone and the results on GOT-10k are obtained without additional training data for fair comparison.

4 Experiments

4.1 Implementation Details

Our experiments are conducted with NVIDIA GeForce RTX 2080 Ti. We adopt ResNet-50 [22] as network backbone which is initialized by the parameters pre-trained on ImageNet-1k [13]. We crop a search patch which is 5^2 times of the target box area from the search frame and resize it to a resolution of 320×320 pixels. The same cropping procedure is also applied to the reference frames. The cropped patches are then down-sampled by the network backbone with a stride of 16. The Transformer encoder consists of 3 layer stacks and the Transformer decoder consists of only 1 layer. The multi-head attention blocks in our tracker have 4 heads with channel width of 256. The inner AiA module reduces the channel dimension of queries and keys to 64. The FFN blocks have 1024 hidden units. Each branch of the target prediction head is comprised of 5 Conv-BN-ReLU layers. The IoU prediction head consists of 3 Conv-BN-ReLU layers, a PrPool [28] layer with pooling size of 3×3 and 2 fully connected layers.

4.2 Results and Comparisons

We compare our tracker with several state-of-the-art trackers on three prevailing large-scale benchmarks (LaSOT [17], TrackingNet and [40] and GOT-10k [25]) and three commonly used small-scale datasets (NfS30 [29], OTB100 [50] and UAV123 [39]). The results are summarized in Tab. 1 and Tab. 2.

LaSOT. LaSOT [17] is a densely annotated large-scale dataset, containing 1400 long-term video sequences. As shown in Tab. 1, our approach outperforms the

Tracker	SiamRPN++ [31]	PrDiMP50 [12]	TransT [8]	STARK-ST50 [52]	KeepTrack [36]	AiATrack (Ours)
NfS30 [29]	50.2	63.5	65.7	65.2	66.4	67.9
OTB100 [50]	69.6	69.6	69.4	68.5	70.9	69.6
UAV123 [39]	61.3	68.0	69.1	69.1	69.7	70.6
Speed (fps)	35	30	50	42	18	38

Table 2. State-of-the-art comparison on commonly used small-scale datasets in terms of AUC score. The best two results are shown in red and blue.

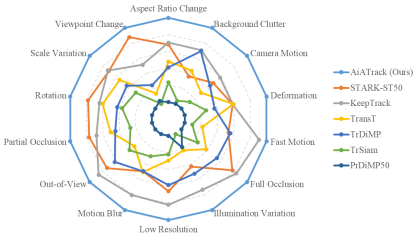

Fig. 5. Attribute-based evaluation on LaSOT in terms of AUC score. Our tracker achieves the best performance on all attribute splits while making a significant improvement in various kinds of scenarios such as background clutter, camera motion, and deformation. Axes of each attribute have been normalized.

previous best tracker KeepTrack [36] by 1.9% in area-under-the-curve (AUC) and 3.6% in precision while running much faster (see Tab. 2). We also provide an attribute-based evaluation in Fig. 5 for further analysis. Our method achieves the best performance on all attribute splits. The results demonstrate the promising potential of our approach for long-term visual tracking.

TrackingNet. TrackingNet [40] is a large-scale short-term tracking benchmark. It provides 511 testing video sequences without publicly available ground truths. Our performance reported in Tab. 1 is obtained from the online evaluation server. Our approach achieve 82.7% in AUC score and 87.8% in normalized precision score, surpassing all previously published trackers. It demonstrates that our approach is also very competitive for short-term tracking scenarios.

GOT-10k. To ensure zero overlaps of object classes between training and testing, we follow the one-shot protocol of GOT-10k [25] and only train our model with the specified subset. The testing ground truths are also withheld and our result is evaluated by the official server. As demonstrated in Tab. 1, our tracker improves all metrics by a large margin, *e.g.* 2.3% in success rate compared with STARK [52] and TrDiMP [48], which indicates that our tracker also has a good generalization ability to the objects of unseen classes.

NfS30. Need for Speed (NfS) [29] is a dataset that contains 100 videos with fast-moving objects. We evaluate the proposed tracker on its commonly used version NfS30. As reported in Tab. 2, our tracker improves the AUC score by 2.7% over STARK [52] and performs the best among the benchmarked trackers.

OTB100. Object Tracking Benchmark (OTB) [50] is a pioneering benchmark for evaluating visual tracking algorithms. However, in recent years, it has been noted that this benchmark has become highly saturated [48,52,36]. Still, the results in Tab. 2 show that our method can achieve comparable performance with state-of-the-art trackers.

UAV123. Finally, we report our results on UAV123 [39] which includes 123 video sequences captured from a low-altitude unmanned aerial vehicle perspective. As shown in Tab. 2, our tracker outperforms KeepTrack [36] by 0.9% and is suitable for UAV tracking scenarios.

4.3 Ablation Studies

To validate the importance of the proposed components in our tracker, we conduct ablation studies on LaSOT testing set and its new extension set [16], totaling 430 diverse videos. We summarize the results in Tab. 3, Tab. 4 and Tab. 5.

Target-Background Embeddings. In our tracking framework, the reference frames not only contain features from target regions but also include a large proportion of features from background regions. We implement three variants of our method to demonstrate the necessity of keeping the context and the importance of the proposed target-background embeddings. As shown in the 1st part of Tab. 3, we start from the variant (a), which is the implementation of the proposed tracking framework with both the target-background embeddings and the AiA module removed. Based on the variant (a), the variant (b) further discards the reference features of background regions with a mask. The variant (c) attaches the target-background embeddings to the reference features. Compared with the variant (a), the performance of the variant (b) drops drastically, which suggests that context is helpful for visual tracking. With the proposed target-background embeddings, the variant (c) can consistently improve the performance over the variant (a) in all metrics. This is because the proposed target-background embeddings further provide cues for distinguishing the target and background regions while preserving the contextual information.

Long-Term and Short-Term Branch. As discussed in Sec. 3.2, it is important to utilize an independent short-term reference branch to deal with the appearance change during tracking. To validate this, we implement a variant (d) by removing the short-term branch from the variant (c). We also implement a variant (e) by adopting a single cross-attention branch instead of the proposed two-branch design for the variant (c). Note that we keep the IoU prediction head for these two variants during training to eliminate the possible effect of IoU prediction on feature representation learning. From the 2nd part of Tab. 3, we can observe that the performance of variant (d) is worse than variant (c), which suggests the necessity of using short-term references. Meanwhile, compared with variant (c), the performance of variant (e) also drops, which validates the necessity to use two separate branches for the long-term and short-term references. This is because the relatively unreliable short-term references may disturb the robust long-term reference and therefore degrade its contribution.

Effectiveness of the AiA Module. We explore several ways of applying the proposed AiA module to the proposed Transformer tracking framework. The variant (f) inserts the AiA module into self-attention blocks in the Transformer encoder. Compared with the variant (c), the performance can be greatly improved on the two subsets of LaSOT. The variant (g) inserts the AiA module into the cross-attention blocks in the Transformer decoder, which also brings a

	Modification		LaSOT [17]			$\text{LaSOT}_{\text{Ext}}$ [16]		
			AUC	P_{Norm}	P	AUC	P_{Norm}	P
1st	(a)	none	65.8	75.8	69.5	44.5	51.5	50.5
	(b)	mask	64.3	72.7	66.6	42.8	50.1	48.8
	(c)	embed†	**67.0**	**77.0**	**71.3**	**44.7**	**52.7**	**51.5**
2nd	(d)	w/o short refer	66.5	76.3	70.7	44.5	51.8	50.6
	(e)	w/o branch split	63.8	72.9	66.7	42.7	50.3	48.6
	(c)	w/ both†	**67.0**	**77.0**	**71.3**	**44.7**	**52.7**	**51.5**
3rd	(c)	w/o AiA†	67.0	77.0	71.3	44.7	52.7	51.5
	(f)	AiA in self-attn	68.6	78.7	72.9	46.2	**54.4**	53.4
	(g)	AiA in cross-attn	67.5	77.9	71.8	46.2	54.2	53.3
	(h)	w/o pos in both	68.0	78.2	72.7	46.2	54.0	53.0
	(i)	AiA in both‡	**68.7**	**79.3**	**73.7**	**46.8**	**54.4**	**54.2**

Table 3. Ablative experiments about different components in the proposed tracker. We use † to denote the basic framework and ‡ to denote our final model with AiA. The best results in each part of the table are marked in **bold**.

consistent improvement. These two variants demonstrate that the AiA module generalizes well to both self-attention blocks and cross-attention blocks. When we apply the AiA module to both self-attention blocks and cross-attention blocks, *i.e.* the final model (i), the performance on the two subsets of LaSOT can be improved by 1.7∼2.7% in all metrics compared with the basic framework (c).

Recall that we introduce positional encoding to the proposed AiA module (see Fig. 2). To verify its importance, we implement a variant (h) by removing positional encoding from the variant (i). We can observe that the performance drops accordingly. This validates the necessity of positional encoding, as it provides spatial cues for consensus seeking in the AiA module. More analysis about the components of the AiA module are provided in the supplementary material.

Superiority of the AiA Module. One may concern that the performance gain of the AiA module is brought by purely adding extra parameters. Thus, we design two other variants to demonstrate the superiority of the proposed module.

First, we implement a variant of our basic framework where each Attention-Add-Norm block is replaced by two cascaded ones. From the comparison of the first two rows in Tab. 4, we can observe that simply increasing the number of attention blocks in our tracking framework does not help much, which demonstrates that our AiA module can further unveil the potential of the tracker.

We also implement a variant of our final model by replacing the proposed inner attention with a convolutional bottleneck [22], which is designed to have a similar computational cost. From the comparison of the last two rows in Tab. 4, we can observe that inserting a convolutional bottleneck can also bring positive effects, which suggests the necessity of correlation refinement. However, the convolutional bottleneck can only perform a fixed aggregation in each local neighborhood, while our AiA module has a global receptive field with dynamic weights determined by the interaction among correlation vectors. As a result, our AiA module can seek consensus more flexibly and further boost the performance.

Modification	Correlation Refinement	LaSOT [17] AUC	P_{Norm}	P	$LaSOT_{Ext}$ [16] AUC	P_{Norm}	P	Speed (fps)
w/o AiA†	✗	67.0	77.0	71.3	44.7	52.7	51.5	44
w/o AiA cascade		67.1	77.0	71.7	44.6	52.9	51.6	40
conv in both	✓	67.9	78.2	72.8	46.0	53.4	52.8	39
AiA in both‡		**68.7**	**79.3**	**73.7**	**46.8**	**54.4**	**54.2**	38

Table 4. Superiority comparison with the tracking performance and the running speed.

Ensemble Size	1	2	3	4	5	6	10
LaSOT [17]	66.8	68.1	68.7	**69.0**	68.2	68.6	68.9
$LaSOT_{Ext}$ [16]	44.9	46.3	46.8	46.2	47.4	**47.7**	47.1
Speed (fps)	39	39	38	38	38	38	34

Table 5. Impact of ensemble size in terms of AUC score and the running speed. All of our ablative experiments are conducted with ensemble size as 3 by default.

Visualization Perspective. In Fig. 4, we visualize correlation maps from the perspective of keys. This is because we consider the correlations of one key with queries as a correlation vector. Thus, the AiA module performs refinement by seeking consensus among the correlation vectors of keys. Actually, refining the correlations from the perspective of queries also works well, achieving 68.5% in AUC score on LaSOT.

Short-Term Reference Ensemble. We also study the impact of the ensemble size in the short-term branch. Tab. 5 shows that by increasing the ensemble size from 1 to 3, the performance can be stably improved. Further increasing the ensemble size does not help much and has little impact on the running speed.

5 Conclusion

In this paper, we present an attention in attention (AiA) module to improve the attention mechanism for Transformer visual tracking. The proposed AiA module can effectively enhance appropriate correlations and suppress erroneous ones by seeking consensus among all correlation vectors. Moreover, we present a streamlined Transformer tracking framework, dubbed AiATrack, by introducing efficient feature reuse and embedding assignment mechanisms to fully utilize temporal references. Extensive experiments demonstrate the superiority of the proposed method. We believe that the proposed AiA module could also be beneficial in other related tasks where the Transformer architecture can be applied to perform feature aggregation and information propagation, such as video object segmentation [53,30,15,34], video object detection [23] and multi-object tracking [46,37,57].

Acknowledgment. This work is supported in part by National Key R&D Program of China No. 2021YFC3340802, National Science Foundation Grant CNS1951952 and National Natural Science Foundation of China Grant 61906119.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 6
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. pp. 850–865. Springer (2016) 1
3. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6182–6191 (2019) 3, 4, 10
4. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Know your surroundings: Exploiting scene information for object tracking. In: European Conference on Computer Vision. pp. 205–221. Springer (2020) 3, 4, 10
5. Bian, J., Lin, W.Y., Matsushita, Y., Yeung, S.K., Nguyen, T.D., Cheng, M.M.: Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4181–4190 (2017) 4
6. Cao, Z., Fu, C., Ye, J., Li, B., Li, Y.: Hift: Hierarchical feature transformer for aerial tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15457–15466 (2021) 1, 3
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020) 3, 4, 6, 7
8. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8126–8135 (2021) 1, 3, 10, 11
9. Cho, S., Hong, S., Jeon, S., Lee, Y., Sohn, K., Kim, S.: Cats: Cost aggregation transformers for visual correspondence. Advances in Neural Information Processing Systems **34** (2021) 4
10. Dai, K., Zhang, Y., Wang, D., Li, J., Lu, H., Yang, X.: High-performance long-term tracking with meta-updater. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6298–6307 (2020) 3
11. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: Accurate tracking by overlap maximization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4660–4669 (2019) 3, 4, 8
12. Danelljan, M., Gool, L.V., Timofte, R.: Probabilistic regression for visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7183–7192 (2020) 10, 11
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 10
14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 3, 4, 6
15. Duke, B., Ahmed, A., Wolf, C., Aarabi, P., Taylor, G.W.: Sstvos: Sparse spatiotemporal transformers for video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5912–5921 (2021) 14

16. Fan, H., Bai, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Huang, M., Liu, J., Xu, Y., et al.: Lasot: A high-quality large-scale single object tracking benchmark. International Journal of Computer Vision **129**(2), 439–461 (2021) 12, 13, 14
17. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5374–5383 (2019) 10, 13, 14
18. Fiaz, M., Mahmood, A., Javed, S., Jung, S.K.: Handcrafted and deep trackers: Recent visual object tracking approaches and trends. ACM Computing Surveys (CSUR) **52**(2), 1–44 (2019) 1
19. Fu, Z., Liu, Q., Fu, Z., Wang, Y.: Stmtrack: Template-free visual tracking with space-time memory networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13774–13783 (2021) 3, 9
20. Ge, W., Lu, X., Shen, J.: Video object segmentation using global and instance embedding learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16836–16845 (2021) 4
21. Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., Shen, C.: Graph attention tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9543–9552 (2021) 3
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 10, 13
23. He, L., Zhou, Q., Li, X., Niu, L., Cheng, G., Li, X., Liu, W., Tong, Y., Ma, L., Zhang, L.: End-to-end video object detection with spatial-temporal transformers. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 1507–1516 (2021) 14
24. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE transactions on pattern analysis and machine intelligence **37**(3), 583–596 (2014) 3
25. Huang, L., Zhao, X., Huang, K.: Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(5), 1562–1577 (2019) 10, 11
26. Huang, L., Wang, W., Chen, J., Wei, X.Y.: Attention on attention for image captioning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4634–4643 (2019) 4
27. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 603–612 (2019) 4
28. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y.: Acquisition of localization confidence for accurate object detection. In: Proceedings of the European conference on computer vision (ECCV). pp. 784–799 (2018) 8, 10
29. Kiani Galoogahi, H., Fagg, A., Huang, C., Ramanan, D., Lucey, S.: Need for speed: A benchmark for higher frame rate object tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1125–1134 (2017) 10, 11
30. Lan, M., Zhang, J., He, F., Zhang, L.: Siamese network with interactive transformer for video object segmentation. arXiv preprint arXiv:2112.13983 (2021) 4, 14
31. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4282–4291 (2019) 1, 10, 11

32. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8971–8980 (2018) 1
33. Li, S., Han, K., Costain, T.W., Howard-Jenkins, H., Prisacariu, V.: Correspondence networks with adaptive neighbourhood consensus. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10196–10205 (2020) 4
34. Mao, Y., Wang, N., Zhou, W., Li, H.: Joint inductive and transductive learning for video object segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9670–9679 (2021) 14
35. Marvasti-Zadeh, S.M., Cheng, L., Ghanei-Yakhdan, H., Kasaei, S.: Deep learning for visual tracking: A comprehensive survey. IEEE Transactions on Intelligent Transportation Systems (2021) 1
36. Mayer, C., Danelljan, M., Paudel, D.P., Van Gool, L.: Learning target candidate association to keep track of what not to track. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13444–13454 (2021) 10, 11, 12
37. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. arXiv preprint arXiv:2101.02702 (2021) 14
38. Min, J., Cho, M.: Convolutional hough matching networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2940–2950 (2021) 4
39. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: European conference on computer vision. pp. 445–461. Springer (2016) 10, 11, 12
40. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 300–317 (2018) 10, 11
41. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 658–666 (2019) 8
42. Rocco, I., Arandjelović, R., Sivic, J.: Efficient neighbourhood consensus networks via submanifold sparse convolutions. In: European conference on computer vision. pp. 605–621. Springer (2020) 4
43. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. Advances in neural information processing systems **31** (2018) 4
44. Sattler, T., Leibe, B., Kobbelt, L.: Scramsac: Improving ransac's efficiency with a spatial consistency filter. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 2090–2097. IEEE (2009) 4
45. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007) 4
46. Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P.: Transtrack: Multiple object tracking with transformer. arXiv preprint arXiv:2012.15460 (2020) 14
47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 1, 2, 3, 4, 7, 8

48. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1571–1580 (2021) 1, 3, 9, 10, 11
49. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018) 4
50. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9), 1834–1848 (2015) 10, 11
51. Xing, D., Evangeliou, N., Tsoukalas, A., Tzes, A.: Siamese transformer pyramid networks for real-time uav tracking. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2139–2148 (2022) 1, 3
52. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10448–10457 (2021) 1, 3, 8, 9, 10, 11
53. Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. Advances in Neural Information Processing Systems **34** (2021) 4, 14
54. Yu, B., Tang, M., Zheng, L., Zhu, G., Wang, J., Feng, H., Feng, X., Lu, H.: High-performance discriminative tracking with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9856–9865 (2021) 1, 3, 10
55. Yu, Y., Xiong, Y., Huang, W., Scott, M.R.: Deformable siamese attention networks for visual object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6728–6737 (2020) 3, 10
56. Zhang, L., Gonzalez-Garcia, A., Weijer, J.v.d., Danelljan, M., Khan, F.S.: Learning the model update for siamese trackers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4010–4019 (2019) 3
57. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864 (2021) 14
58. Zhang, Z., Peng, H., Fu, J., Li, B., Hu, W.: Ocean: Object-aware anchor-free tracking. In: European Conference on Computer Vision. pp. 771–787. Springer (2020) 1, 10
59. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020) 3, 4