Compact Transformer Tracker with Correlative Masked Modeling

Zikai Song[1], Run Luo[1], Junqing Yu[1*], Yi-Ping Phoebe Chen[2], Wei Yang[1*]

[1]Huazhong University of Science and Technology, China
[2]La Trobe University, Australia
{skyesong, lr_8823, yjqing, weiyangcs}@hust.edu.cn, phoebe.chen@latrobe.edu.au

Abstract

Transformer framework has been showing superior performances in visual object tracking for its great strength in information aggregation across the template and search image with the well-known attention mechanism. Most recent advances focus on exploring attention mechanism variants for better information aggregation. We find these schemes are equivalent to or even just a subset of the basic self-attention mechanism. In this paper, we prove that the vanilla self-attention structure is sufficient for information aggregation, and structural adaption is unnecessary. The key is not the attention structure, but how to extract the discriminative feature for tracking and enhance the communication between the target and search image. Based on this finding, we adopt the basic vision transformer (ViT) architecture as our main tracker and concatenate the template and search image for feature embedding. To guide the encoder to capture the invariant feature for tracking, we attach a lightweight correlative masked decoder which reconstructs the original template and search image from the corresponding masked tokens. The correlative masked decoder serves as a plugin for the compact transformer tracker and is skipped in inference. Our compact tracker uses the most simple structure which only consists of a ViT backbone and a box head, and can run at 40 *fps*. Extensive experiments show the proposed compact transform tracker outperforms existing approaches, including advanced attention variants, and demonstrates the sufficiency of self-attention in tracking tasks. Our method achieves state-of-the-art performance on five challenging datasets, along with the VOT2020, UAV123, LaSOT, TrackingNet, and GOT-10k benchmarks. Our project is available at https://github.com/HUSTDML/CTTrack.

1 Introduction

Visual Object Tracking is one of the fundamental tasks in computer vision with applications ranging from human-computer interaction, surveillance, traffic flow monitoring and etc. It aims to estimate the location, denoted as a bounding box, of an arbitrary target object throughout the subsequent video sequence. Deep Learning based trackers have achieved great success due to their strong representation ability. Trackers (Bertinetto et al. 2016; Nam and Han 2016;

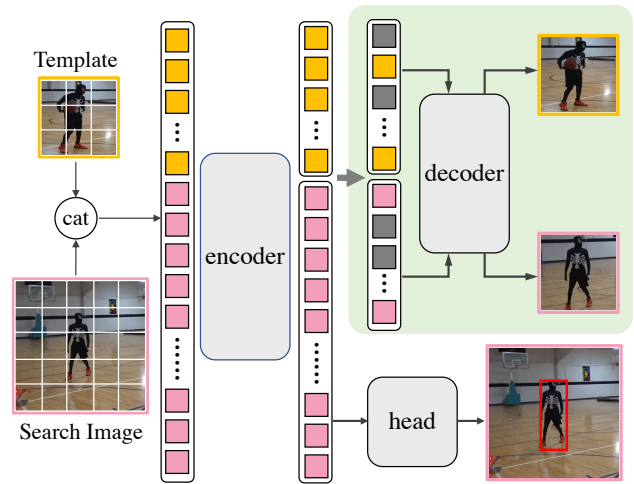---

*indicates co-corresponding author.

Figure 1: Our compact transformer tracker adopts the simple ViT structure (encoder) with the concatenation of the template and search image as input, which essentially exploits the standard self-attention mechanism for information aggregation. The encoded tokens pass through a box head to estimate the result bounding box. And we develop a correlative masked decoder reconstructing the original template and search pixels to enhance the information aggregation, which is skipped during inference.

Li et al. 2018, 2019) derived from Convolutional Neural Networks (CNN) (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2015; He et al. 2016) produce tracking accuracy that beyond the comparison of traditional approaches, especially the trackers built on Siamese network (Bertinetto et al. 2016; Xu et al. 2020; Li et al. 2018, 2019; Voigtlaender et al. 2020; Yu et al. 2020; Guo et al. 2021). The key of Siamese network trackers is to produce the cross-correlation and measure the similarity between the target template and search image. Nowadays, transformer-based trackers (Chen et al. 2021; Wang et al. 2021; Yan et al. 2021; Shen et al. 2022; Song et al. 2022; Cui et al. 2022) have shown great strength by introducing the attention mechanism (Vaswani et al. 2017) to enhance and fuse the features of querying sample and tracked objects. Prevalent transformer trackers (Chen et al. 2021; Yan et al. 2021;

Cui et al. 2022) more or less adapt the attention for aggregating information across the template and search image.

We find that the advanced variants of attention mechanism in recent research, including mix-attention (Cui et al. 2022) and cross-attention (Yu et al. 2020; Chen et al. 2021), are equivalent or even just a subset of the packed self-attention (i.e., standard self-attention with the concatenation of the template and search image as input). Then the question is which parts of the self-attention mechanism play an important role in visual object tracking? We revisited the transformer tracking framework and find that the tracking results are generated from tokens corresponding to the search image (search tokens), while the tokens corresponding to the template (template tokens) are always discarded in the last. The representational ability of search tokens comes from two parts: the cross-information enhancement from the template tokens and the self-information enhancement from the search tokens themselves. In this paper, we prove that self-information enhancement in multi-image attention plays a greater role than cross-information aggregation, though cross-information aggregation is indispensable in visual object tracking but not greatly beneficial.

Driven by this analysis, we propose a compact transformer tracker combined with correlative masked modeling for the cross-information aggregation and self-information reinforcement. As shown in Figure 1, our tracker adopts the basic vision transformer as the main branch and applies a lightweight masked decoder to enhance the implicit representation capability of the packed self-attention. The correlative masked decoder, which is inspired by Masked Image Modeling (He et al. 2022; Xie et al. 2022), reconstructs the both original template and search pixels from the corresponding masked tokens, to guide the encoder to capture the invariant feature for tracking. In addition, our decoder can be plugged into other transformer trackers, which can effectively improve the tracking performance without compromising speed. Applying our correlative masked modeling strategy to the compact transformer tracker can improve the AUC from 64.0% to 65.8% on the LaSOT (Fan et al. 2019) dataset. Extensive comparison experiments on 5 challenging datasets including VOT2020 (Kristan et al. 2020), UAV123 (Mueller, Smith, and Ghanem 2016), LaSOT, GOT-10k (Huang, Zhao, and Huang 2019), and TrackingNet (Muller et al. 2018) exhibits the state-of-the-art performance, which further evidence the correctness of our analysis regarding the self-attention in visual tracking.

To summarize, our main contributions include:

1. We present a unified analyzing method for the attention mechanism and find that the advanced variants of the attention mechanism are equivalent or even just a subset of the self-attention. We also prove that self-information enhancement in multi-image attention plays a greater role than cross-information aggregation.

2. We develop a compact transformer tracker with a correlative masked decoder, which has a very simple structure and achieves state-of-the-art accuracy at a high Frames-Per-Seconds (*fps*) tracking speed. The decoder reconstructs the original template and search image from the

corresponding masked tokens and serves as a training plugin for the tracker. The experiment demonstrates that our analysis regarding self-attention is correct.

2 Related Work

Traditional trackers. Traditional single object tracking algorithms can be roughly summarized as Correlation Filter based trackers (CF), Deep Network based trackers (DLN). CF-based trackers(Bolme et al. 2010; Henriques et al. 2015; Danelljan et al. 2016, 2017, 2019; Bhat et al. 2019) exploit the convolution theorem and learn a filter in the Fourier domain that maps known target images to the desired output. DLN-based trackers refer to algorithms employing deep neural networks for the tracking process. Earlier approaches (Nam and Han 2016; Pu et al. 2018) treat the tracking task as a classification problem and exploit deep features for locating the target. Shortly afterwards more trackers adopt the Siamese network (Bertinetto et al. 2016; Li et al. 2018, 2019) for its effectiveness in measuring similarity. The Siamese network consists of two branches, one operates on the template and the other for the search area.

Above all, these methods mainly consist of a backbone which extracts the features of search image and template separately, a similarity measuring module, and heads to predict the location and bounding box. Compared to our framework, traditional trackers have too many modules and a very complex design, we simply adapt a ViT backbone with a box head to get better tracking results.

Transformer trackers. The ViT (Dosovitskiy et al. 2021) first introduces the transformer to image recognition tasks and presents an impressive performance. Ever since, transformer has been widely applied in image classification(Dosovitskiy et al. 2021; Wu et al. 2021; Liu et al. 2021), object detection(Carion et al. 2020; Li et al. 2022), visual object tracking(Yan et al. 2021; Chen et al. 2021; Wang et al. 2021; Song et al. 2022; Shen et al. 2022; Cui et al. 2022) and etc. Transformer-based tracking methods have become the mainstream tracking algorithms nowadays. TransT (Chen et al. 2021) proposes a feature fusion network and employs an attention mechanism to combine the features of the template and search region. STARK (Yan et al. 2021) develops a spatial-temporal architecture based on the encoder-decoder transformer. CSWinTT (Song et al. 2022) proposes a transformer architecture with multi-scale cyclic shifting window attention for visual tracking, elevating the attention from pixel level to window level. MixFormer (Cui et al. 2022) constructs a compact tracking framework and designs a mixed attention module that unifies the process of feature extraction and information matching module.

Instead of designing a complex attention mechanism as in the previous tracking approaches, we compare the essential differences of attention variants(such as mix-attention and cross-attention) and find these attention variants are equivalent or even just a subset of the packed self-attention. To verify the capability of self-attention in information aggregation, we design a compact transformer tracker using the most simple pipeline which only consists of a ViT backbone and a box head, without any extra design including separate

modules of feature extraction and aggregation, and multi-layer feature aggregation.

Masked image modeling (MIM). MIM masks an area of the original images and predicts the missing pixels, which aims to enhance the representation of models. Recently, MIM approaches((Chen et al. 2020; He et al. 2022; Xie et al. 2022; Wei et al. 2022; Bao, Dong, and Wei 2021)) are extended to the modern vision transformers (Dosovitskiy et al. 2021; Liu et al. 2021). iGPT (Chen et al. 2020) first proposes a transformer to predict unknown pixels from a sequence of low-resolution pixels. BEiT (Bao, Dong, and Wei 2021) tokenizes the images via an additional dVAE (Ramesh et al. 2021) network with a block-wise masking strategy. Sim-MIM (Xie et al. 2022) find that a moderately large masked patch size of the input image for pixel predictions makes a strong pre-text task. MAE (He et al. 2022) develops an asymmetric encoder-decoder architecture, the encoder operates on a small proportion of the visible patches, and the decoder reconstructs the original pixels. MaskFeat (Wei et al. 2022) reconstructs the feature descriptors such as HoG (Dalal and Triggs 2005) instead of pixels.

Our approach is inspired by the previous MIM method (Xie et al. 2022; He et al. 2022), but we have to deal with two fundamental problems in the tracking framework: (1) Visual tracking is a downstream vision task that generally does not have the pre-train process to apply the MIM strategy. We develop a masked decoder to leverage the search and the template tokens to predict the original images, which is embedded as an attachment plugin in the training phase to implement an end-to-end model. (2) MIM methods reconstructing the single image do not fit the tracking framework which involves cross-aggregation of multiple images. According to the properties of packed self-attention, we design a self-decoder and a cross-decoder to reconstruct the original template and search image from the corresponding masked tokens. As far as we know, we are the first to artfully introduce the MIM into the visual tracking field to improve the information aggregation capabilities.

3 Approach

In this section, we introduce our compact transformer tracker with correlative masked modeling in detail. Before proceed, we first present a analysis on the key component of transformer tracker, and demonstrate that existing attention variants are equivalent to the packed self-attention.

3.1 Revisiting Transformer Tracker

Transformer tracking framework. As described in ViT(Vaswani et al. 2017), the query-key-value attention mechanism is applied with query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$. The linear weights of $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$ respectively. The attention (Attn) is computed as:

$$\text{Attn}(\mathbf{X}) = \text{softmax}(\frac{\mathbf{X}\mathbf{W}_Q \cdot \mathbf{W}_K^T\mathbf{X}^T}{\sqrt{d_k}}) \cdot \mathbf{X}\mathbf{W}_V \quad (1)$$

where the X is the input token and the d_k is the dimension of the key. For a clearer description of the post-order steps,
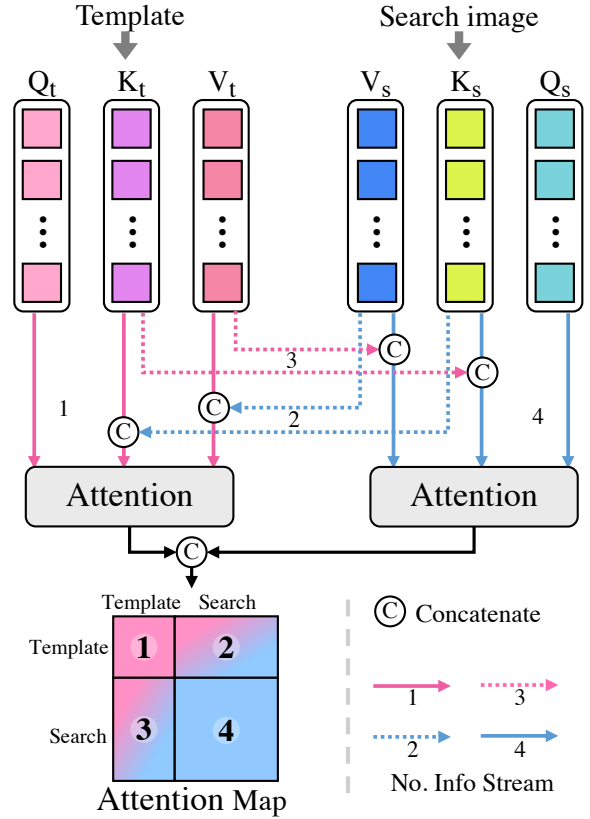

Figure 2: Information streams in the attention mechanism. The four information streams of Q-K-V are corresponding to the four parts in the attention map. Variants of attention can be uniformly explained under this analytical approach.

we apply an attention calculation with the inputs of two different tokens, the token $\mathbf{X}_Q$ computed with query and the token $\mathbf{X}_{KV}$ computed with key and value. We modify the attention formula and define the attention map (AMap) as:

$$\text{Attn}(\mathbf{X}_Q, \mathbf{X}_{KV}) = \text{AMap}(\mathbf{X}_Q, \mathbf{X}_{KV}) \cdot \mathbf{X}_{KV}\mathbf{W}_V$$
$$\text{AMap}(\mathbf{X}_Q, \mathbf{X}_{KV}) = \text{softmax}(\frac{\mathbf{X}_Q\mathbf{W}_Q \cdot \mathbf{W}_K^T\mathbf{X}_{KV}^T}{\sqrt{d}}) \quad (2)$$

Our compact transformer tracker consists of two parts: a transformer backbone for information aggregation and a box head for the bounding box estimation. Give the template z in the initial frame and a search image s. We obtain the tokens $X_t \in \mathbb{R}^{L_z \times d}$ and $X_s \in \mathbb{R}^{L_s \times d}$ respectively through patch embedding, where d represents the number of channels. The **packed self-attention (PSelf-Attn)** in the tracking field is defined as the self-attention with the input of the concatenation (Cat) of the template and the search image:

$$\text{PSelf-Attn} = \text{Attn}\Big(Cat(\mathbf{X}_z, \mathbf{X}_s), Cat(\mathbf{X}_z, \mathbf{X}_s)\Big) \quad (3)$$

Analysis on Attention. As shown in Figure 2, we divide the computation of attention mechanism, which involves both template and search image, into four information streams:
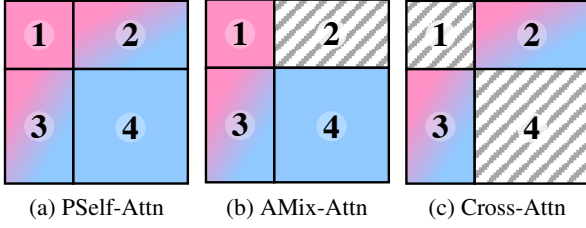
(a) PSelf-Attn (b) AMix-Attn (c) Cross-Attn

Figure 3: Configurations of information stream in attention map of packed self-attention (PSelf-Attn), asymmetric mix-attention(AMix-Attn) and cross-attention (Cross-Attn).

(1) self-information enhancement on template;

(2) cross-information aggregation on template;

(3) cross-information aggregation on search image;

(4) self-information enhancement on search image.

These four information streams are also reflected in the four parts of the attention map (In Figure 2, the index of each part in the attention map corresponds to the information stream). Based on this dissection, we can conveniently compare the differences between existing attention, including packed self-attention, mix-attention, and cross-attention.

The **PSelf-Attn** and the **mix-attention**(Cui et al. 2022) are essentially equivalent, the mix-attention is calculated as:

$$\text{PSelf-Attn} == \text{Mix-Attn} =$$
$$\text{Cat}\Big(\text{AMap}\big(\mathbf{X}_z, Cat(\mathbf{X}_z, \mathbf{X}_s)\big), \text{AMap}\big(\mathbf{X}_s, Cat(\mathbf{X}_z, \mathbf{X}_s)\big)\Big) \quad (4)$$

which is the same as Eqn. 3, and they include all four information streams (the attention map is shown as Figure 3a).

By the same analysis, the **asymmetric mix-attention (AMix-Attn)** contains three information streams (#1, #3, #4 info stream), which is shown in the Figure 3b and is calculated as follows:

$$\text{AMix-Attn} =$$
$$\text{Cat}\Big(\text{AMap}\big(\mathbf{X}_z, \mathbf{X}_z\big), \text{AMap}\big(\mathbf{X}_s, Cat(\mathbf{X}_z, \mathbf{X}_s)\big)\Big) \quad (5)$$

The **cross-attention** contains two information streams (#2,#3 info stream) for cross information aggregation, which is shown in the Figure 3c and is calculated as follows:

$$\text{Cross-Attn} = \text{Cat}\Big(\text{AMap}\big(\mathbf{X}_z, \mathbf{X}_s\big), \text{AMap}\big(\mathbf{X}_s, \mathbf{X}_z\big)\Big) \quad (6)$$

In order to fully verify the importance of each part of packed attention, it is necessary to evaluate the impact of each information stream individually. The key of visual object tracking is to find the target in the search image, there must be a cross-information aggregation of the search image (#3 info stream). The other information streams can be blocked out to verify their performance.

Based on the above idea, we conduct detailed experiments and the result is shown in Table 1. Removing cross-information aggregation of the template (#2 info stream) of

# AMap	No. Info Stream				AUC	Prec
	①	②	③	④		
1	✓	✓	✓	✓	61.7	64.2
2	✓		✓	✓	**64.0**	**67.7**
3		✓	✓	✓	60.6	63.7
4	✓	✓	✓		58.8	60.1
5		✓	✓		57.9	58.5

Table 1: The effectiveness of information streams in the attention mechanism on the LaSOT dataset. The visualized four parts in the attention map (AMap) correspond to the four information streams at the matched location.

self-attention can greatly improve tracking performance (the AUC and Prec of Table 1 #2 are better than that of Table 1 #1), and the cross-information aggregation of the template will introduce a lot of noise in template features, which is not recommended in visual tracking. However, removing self-information enhancement (#3 and #4 info stream) of self-attention severely degrades the tracking performance (the AUC and Prec of Table 1 #3 and #4 are worse than that of Table 1 #1). From the results we can conclude that self-information enhancement in multi-image attention plays a greater role than cross-information aggregation, the cross-information aggregation is indispensable in tracking but not greatly beneficial.

3.2 Correlative Masked Modeling

According to the above analysis, the best tracking performance can be achieved by adopting three information streams: self-information on the template(#1 info stream), cross-information on the search image (#3 info stream), and self-information on the search image (#4 info stream). These three information streams can be grouped into two categories: two self-information enhancements and one cross-information aggregation. We designed a correlative masked modeling method to enhance the information aggregation of our tracking framework, as shown in Figure 1. The ViT backbone is an encoder, and the correlative masked decoder reconstructs the original image (the template and search image respectively) from randomly masked tokens to enhance the self-information and reconstructs the template image from search tokens to improve cross-information aggregation. In parallel with the masked decoder, the search image tokens go through a box estimation head as in (Yan et al. 2021) to generate the result bounding box.

Decoder. The decoders in our framework consist of a self-decoder and a cross-decoder, these two decoders have the same structure but do not share weights, each one is composed of a series of transformer blocks similar to the MAE, and the last layer of the decoder is a linear projection with output channels equal to the number of pixels in a patch. As shown in Figure 4, the decoder takes masked tokens as input and predicts the original image pixels corresponding to
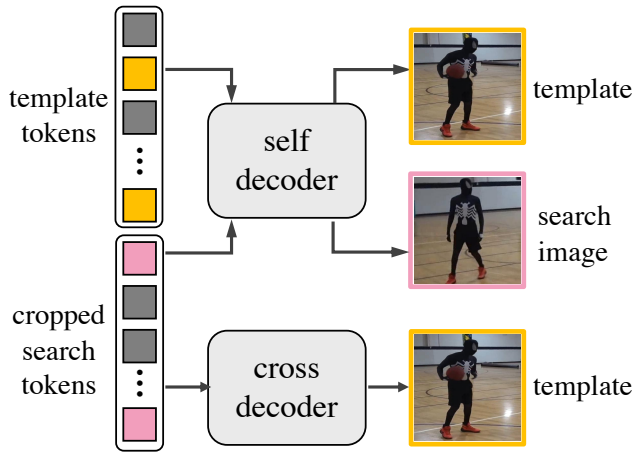
Figure 4: The correlative masked decoders consists of a self-decoder and a cross-decoder. The self-decoder reconstructs the two original images, template and search image, from its corresponding masked tokens. The cross-decoder reconstructs the template image from search tokens.

the template token and the search image token, where the template tokens are only self-reconstructed to the template image for enhancing the #1 information stream, search tokens are used to crossly reconstruct the template image (for #3 info stream) and self-reconstruct the search image (for #4 info stream).

Masking and Reconstruction. The encoder embeds the concatenation set of template tokens and search tokens. Then we split the encoded tokens into template tokens and search tokens, crop the search tokens using Precise RoI Pooling(Jiang et al. 2018) to the same size as the template tokens, and sample a subset of them. We randomly sample tokens at a high masking ratio (75%). Our decoder predicts the pixel values for each masked token, and the output of the decoder is reshaped to form a reconstructed image. We use the mean squared error (MSE) between the reconstructed and original images on masked tokens as our loss function.

3.3 Training and Inference

Our decoder is only used in the training phase, while does not participate in the inference phase, hence it doesn't affect the tracking speed. During the training phase, our tracker takes a triplet input consisting of one search region and two templates similar to STARK(Yan et al. 2021). We randomly sample multiple frames from sequences in the training set, select the first frame and the second frame as templates, and the last frame as the search region. In the target localization training, we train the whole network except the scoring head in an end-to-end manner with the combination of $L1$ Loss, generalized IoU loss (Rezatofighi et al. 2019), and decoder loss L_{dec}. The full loss function is defined as follows:

$$Loss = \lambda_{L1} L_1(B_i, \hat{B}_i) + \lambda_g L_g(B_i, \hat{B}_i) + \lambda_{dec} L_{dec} \quad (7)$$

where $\lambda_{L1} = 5.0$, $\lambda_g = 2.0$ and $\lambda_{dec} = 0.3$ are the weighting factors of three losses, $\hat{B}_i$ is the estimated box of the target and B_i is the ground-truth bounding box. The decoder

loss L_{dec} is defined as:

$$L_{dec} = L_2(z, z_p) + L_2(s, s_p) + L_2(z, s_p) \quad (8)$$

where the L_2 is the MSE loss, z and s represent the original template image and search image, z_p and s_p represent the predicting template image and search image respectively.

In the inference phase, we use two templates of the same size as the input. One of which is the initial template and fixed, the other is online updated and always set to the latest tracking result with high confidence. We use a score head to control the updating of the online template. Our score head consists of the multilayer perceptron (MLP) that receives a class-token(Dosovitskiy et al. 2021) as input and evaluates the accuracy of current tracking results.

4 Experiments

4.1 Implementation Details

In order to effectively verify the correctness of our analysis, we design the compact transformer tracker without any other extra attention mechanisms. The only structures remaining are feature extraction and aggregation, and multi-layer feature aggregation. The main tracker only consists of a ViT backbone and a box estimation head, we test both ViT-Base and ViT-Large, and the ViT parameters are initialized with MAE (He et al. 2022) pre-trained model. We refer our **C**ompact **T**ransformer tracker as CTTrack-B (the backbone of ViT-Base) and CTTrack-L (the backbone of ViT-Large) in this section.

We adopt CoCo(Lin et al. 2014), LaSOT(Fan et al. 2019), GOT-10k(Huang, Zhao, and Huang 2019), and TrackingNet(Muller et al. 2018) as our training dataset except the GOT-10k benchmark. The training samples are directly sampled from the same sequence and we apply common data augmentation operations including brightness jitter and horizontal flip. The size of the input template is 128×128, the search region is 5^2 times of the target box area and further resized to 320×320. The decoder parameters are initialized with Xavier Uniform. The AdamW optimizer (Loshchilov and Hutter 2018) is employed with initial learning rate (lr) of 1e-4 with the layer-wise decay 0.75, and the lr decreases according to the cosine function with the final decrease factor of 0.1. We adopt a warm-up lr with the 0.2 warm-up factor on the first 5 epochs. We train our model on 4 Nvidia Tesla V100 GPUs for a total of 500 epochs, each epoch uses $6×10^4$ images. The mini-batch size is set to 128 images with each GPU hosting 32 images. Our approach is implemented in Python 3.7 with PyTorch 1.7.

4.2 Ablation Study

We ablate our compact transformer tracker on several intriguing properties using the challenging LaSOT dataset and report the Area Under the Curve (AUC) and Precision (Prec) as the validation accuracy.

Backbone Comparison. Table 2 shows the comparison of the transformer backbones between the ViT-Base and ViT-Large backbone. The CTTrack-B reaches a higher tracking speed while the CTTrack-L exhibits a better performance.

Methods	Params(M)	FLOPs(G)	Speed(*fps*)
CTTrack-B	93.8	48.1	40
CTTrack-L	313.9	163.7	22

Table 2: Model size and speed using different backbones.

Reconstruction Streams. Our decoder enforces three types of reconstruction streams as shown in Figure 4. Table 3 exhibits different configurations of reconstruction streams, through varied combinations of search tokens reconstruct search image (s2s), template tokens reconstruct template image (t2t) and search tokens reconstruct template image(s2t). The result is consistent with the conclusion of our previous analysis that self-information enhancement (#5) plays the most important role in transformer tracking, compared to cross-information aggregation(#4). Besides, search image information has more influence than the template information, the s2s (#2) improves performance the most among all streams (#2, #3, #4), from 64.0 to 64.7 in AUC score. After adopting all three reconstruction streams, tracking accuracy improved by an impressive AUC score of 1.8%, which validates the effectiveness of our masked modeling decoders.

#	Recons Type			AUC	Prec
	s2s	t2t	s2t		
1	-	-	-	64.0	67.7
2	✓	-	-	64.7	69.1
3	-	✓	-	64.4	68.4
4	-	-	✓	64.4	68.6
5	✓	✓	-	65.1	69.9
6	✓	✓	✓	**65.8**	**70.9**

Table 3: Ablation Study for the reconstruction streams. The s2s represents search tokens reconstruct search image, t2t denotes template tokens reconstruct template image and s2t means search tokens reconstruct template image.

Masking ratio. When we conduct reconstruction streams, we randomly mask the input tokens according to a predefined ratio. Table 4 shows the influence of different masking ratios. We mask the encoded template token and search tokens with a random sampling strategy at different masking rates. Similar to the conclusion obtained by the MAE(He et al. 2022), the optimal ratios are relatively high, and the accuracy increases steadily with the masking ratio growing until reaching 75%, which produces the best tracking results.

Mask Ratio	25%	50%	75%	90%
AUC	64.6	65.7	**65.8**	64.9
Prec	69.0	70.7	**70.9**	69.5

Table 4: Comparison on masking ratio.

Online Template Updating. We evaluate the effect of the online update strategy in our method. The ablation study
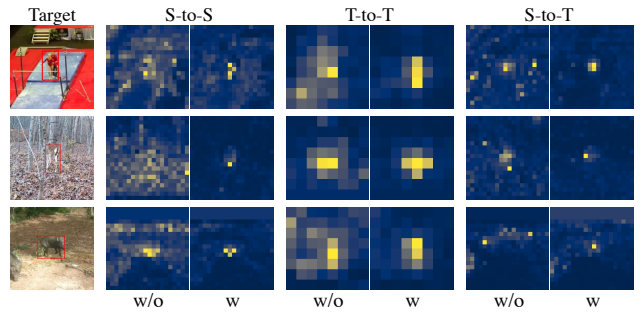

Figure 5: Visualization of attention map which compares the difference between training with correlative decoder (w) and training without correlative decoder(w/o). S-to-S is self-information enhancement on search image, T-to-T is self-information enhancement on template, S-to-T is cross-information aggregation on search image.

result is shown in Table 5, #1 represents the performance without template updating. We can see that applying a fixed interval to update the online template (#2) is ineffective as it greatly reduces the quality of template and causes tracking drift. It can be seen in #3, there is a 0.2% improvement in the AUC score after applying the scoring head to evaluate the accuracy of current tracking results.

	Online	Score	AUC	Prec
	-	-	65.8	70.9
CTTrack-B	✓	-	64.9	69.9
	✓	✓	66.0	71.1

Table 5: Ablation for the online template updating component. Online denotes updating the template at a fixed update interval. Score represents the online template is only updated with high confident samples.

Visualization of attention maps. We visualize attention maps in Figure5, our tracker adopting the correlative decoder has a stronger discriminative ability. The baseline transformer without a reconstruction decoder tends to lose the target position, and the distractors in the background get suppressed with the training by the correlative decoder.

4.3 Comparison with the SOTA

We compare our compact tracker with the state-of-the-art trackers on UAV123(Mueller, Smith, and Ghanem 2016), LaSOT(Fan et al. 2019), TrackingNet(Muller et al. 2018), GOT-10k(Huang, Zhao, and Huang 2019), and VOT2020(Kristan et al. 2020). For a fairer comparison, here we adopt relative position biases in our ViT backbones, this addition improves AUC by around 1 point.
UAV123 gathers an application-specific collection of 123 sequences. It adopts the AUC and Precision (P) as the evaluation metrics. As shown in Table 1, Our CTTrack-L outperforms previous trackers and exhibits very competitive performance (71.3% AUC) when compared to the previous best-performing tracker CSWinTT (70.5% AUC).

Methods	UAV123		LaSOT			TrackingNet			GOT-10k		
	AUC	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
CTTrack-L	**71.3**	**93.3**	*69.8*	*79.7*	*76.2*	**84.9**	**89.1**	**83.5**	75.3	84.5	74.0
CTTrack-B	68.8	89.5	67.8	77.8	74.0	82.5	87.1	80.3	73.5	83.5	70.6
CTTrack-L -GOT	-	-	-	-	-	-	-	-	**72.8**	**81.3**	**71.5**
CTTrack-B -GOT	-	-	-	-	-	-	-	-	*71.3*	*80.7*	*70.3*
MixFormer(Cui et al. 2022)	69.5	*91.0*	**70.1**	**79.9**	**76.3**	*83.9*	*88.9*	*83.1*	70.7	80.0	67.8
CSWinTT(Song et al. 2022)	*70.5*	90.3	66.2	75.2	70.9	81.9	86.7	79.5	69.4	78.9	65.4
UTT(Shen et al. 2022)	-	-	64.6	-	67.2	79.7	-	77.0	67.2	76.3	60.5
STARK(Yan et al. 2021)	-	-	67.1	77.0	-	82.0	86.9	-	68.8	78.1	64.1
TransT(Chen et al. 2021)	68.1	87.6	64.9	73.8	69.0	81.4	86.7	80.3	67.1	76.8	60.9
TrDiMP(Wang et al. 2021)	67.0	87.6	64.0	73.2	66.6	78.4	83.3	73.1	68.8	80.5	59.7
STMTrack(Fu et al. 2021)	64.7	-	60.6	69.3	63.3	80.3	85.1	76.7	64.2	73.7	57.5
AutoMatch(Zhang et al. 2021)	64.4	83.8	58.2	67.5	59.9	76.0	82.4	72.5	65.2	76.6	54.3
SiamGAT(Guo et al. 2021)	64.6	84.3	53.9	63.3	53.0	-	-	-	62.7	74.3	48.8
KYS(Bhat et al. 2020)	-	-	55.4	63.3	55.8	74.0	80.0	68.8	63.6	75.1	51.5
MAML(Wang et al. 2020)	-	-	52.3	-	53.1	75.7	82.2	72.5	-	-	-
SiamAttn(Yu et al. 2020)	65.0	84.5	56.0	64.8	-	75.2	81.7	-	-	-	-
SiamFC++(Xu et al. 2020)	61.8	80.4	54.4	62.3	54.7	75.4	80.0	70.5	59.5	69.5	47.9
SiamRPN++(Li et al. 2019)	64.2	84.0	49.6	56.9	49.1	73.3	80.0	69.4	51.7	61.6	32.5
DiMP(Bhat et al. 2019)	64.2	84.9	57.7	66.4	57.9	74.0	80.1	68.7	61.1	71.7	49.2
ATOM(Danelljan et al. 2019)	61.7	82.7	51.5	57.6	50.5	70.3	77.1	64.8	55.6	63.4	40.2

Table 6: Comparisons with previous state-of-the-art trackers on four challenge benchmarks. The bold and italic performances ranked at first and second places. The GOT denotes only trained on the GOT-10k train split.

Methods	EAO↑	Accuracy↑	Robustness↑
SiamFC	0.179	0.418	0.502
ATOM	0.271	0.462	0.734
DiMP	0.274	0.457	0.740
UPDT	0.278	0.465	0.755
TransT	*0.293*	*0.477*	0.754
CSWinTT	**0.304**	**0.480**	**0.787**
CTTrack-L	0.287	0.453	**0.787**

Table 7: Comparisons on VOT2020, where trackers only predict bounding boxes rather than masks.

LaSOT is a long-term dataset including 1400 sequences and distributed over 14 attributes, the testing subset of La-SOT contains 280 sequences. Methods are ranked by the AUC, P, and Normalized Precision (P_{Norm}). Our CTTrack-L achieves the AUC (69.8%) and Prec (76.2%), which is an excellent result that outperforms other methods only except the MixFormer. Our tracker has lower performance than MixFormer on LaSOT because it contains long-term sequences and large variations in content. ViT backbone is a plain and non-hierarchical architecture that maintains feature maps at a certain scale, which may not be able to well handle long-term tracking sequences with scale variations.

TrackingNet is a large-scale tracking dataset consisting of 511 sequences for testing. The evaluation is performed on the online server. Table 1 shows that CTTrack-L performs better quality and ranks first in AUC score at 84.9%. The gain is 1.0% improvement when compared with the previous best results.

GOT-10k contains over 10k videos for training and 180 for testing. It forbids the trackers to use external datasets for training. We follow this protocol by retraining our trackers to only use the GOT10k train split. As in Table 1, MixFormer and CSWinTT provide the best performance, with an AO score of 70.7% and 69.4%. Our CTTrack-L has obtained an AO score of 72.8%, significantly outperforming the best existing tracker by 2.1%.

VOT2020 benchmark contains 60 challenging videos. The performance is evaluated using the expected average overlap (EAO), which takes both accuracy (A) and robustness (R). Since our algorithm does not output a segmentation mask, trackers that only predict bounding boxes are selected for comparisons to ensure fairness. It can be seen from Table 7 that our CTTrack-L obtains an EAO of 0.287.

5 Conclusion

In this work, we analyze the information stream in the attention mechanism in depth. We prove that the vanilla self-attention structure is sufficient for information aggregation, and employ the three information streams of the packed self-attention in the transformer tracking framework. To enhance the information representation, we design the correlative masked decoder consisting of a self-decoder and a cross-decoder to reconstruct the original pixels of both template and search image. Extensive experiments demonstrate the effectiveness of our correlative masked modeling strategy and our compact transformer tracker exhibits impressive performance over previous trackers. In addition, our correlative masked decoder can be plugged into other transformer trackers, which can effectively improve the tracking performance without compromising speed. In the future, we plan to combine the feature pyramid or convolution module for better performance on long-term tracking sequences.

Acknowledgments

References

Bao, H.; Dong, L.; and Wei, F. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. S. 2016. Fully-Convolutional Siamese Networks for Object Tracking. In *Proceedings of the ECCV*, 850–865. Springer.

Bhat, G.; Danelljan, M.; Gool, L. V.; and Timofte, R. 2019. Learning Discriminative Model Prediction for Tracking. In *Proceedings of the ICCV*, 6182–6191. IEEE.

Bhat, G.; Danelljan, M.; Van Gool, L.; and Timofte, R. 2020. Know Your Surroundings: Exploiting Scene Information for Object Tracking. In *Proceedings of the ECCV*. Springer.

Bolme, D. S.; Beveridge, J. R.; Draper, B. A.; and Lui, Y. M. 2010. Visual object tracking using adaptive correlation filters. In *Proceedings of the CVPR*, 2544–2550. IEEE.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*, 213–229. Springer.

Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; and Sutskever, I. 2020. Generative pretraining from pixels. In *International conference on machine learning*, 1691–1703. PMLR.

Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; and Lu, H. 2021. Transformer tracking. In *Proceedings of the CVPR*, 8126–8135.

Cui, Y.; Jiang, C.; Wang, L.; and Wu, G. 2022. MixFormer: End-to-End Tracking With Iterative Mixed Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13608–13618.

Dalal, N.; and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, 886–893. Ieee.

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

Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; and Felsberg, M. 2017. ECO: Efficient Convolution Operators for Tracking. In *Proceedings of the CVPR*, 6638–6646. IEEE.

Danelljan, M.; Robinson, A.; Khan, F. S.; and Felsberg, M. 2016. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In *Proceedings of the ECCV*, 472–488. Springer.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.;

Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*.

Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; and Ling, H. 2019. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In *Proceedings of the CVPR*. IEEE.

Fu, Z.; Liu, Q.; Fu, Z.; and Wang, Y. 2021. Stmtrack: Template-free visual tracking with space-time memory networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13774–13783.

Guo, D.; Shao, Y.; Cui, Y.; Wang, Z.; Zhang, L.; and Shen, C. 2021. Graph attention tracking. In *Proceedings of the CVPR*, 9543–9552.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16000–16009.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the CVPR*, 770–778. IEEE.

Henriques, J. F.; Caseiro, R.; Martins, P.; and Batista, J. 2015. High-Speed Tracking with Kernelized Correlation Filters. *IEEE TPAMI*, 37(3): 583–596.

Huang, L.; Zhao, X.; and Huang, K. 2019. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. *IEEE TPAMI*.

Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; and Jiang, Y. 2018. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European conference on computer vision (ECCV)*, 784–799.

Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.-K.; Danelljan, M.; Zajc, L. Č.; Lukežič, A.; Drbohlav, O.; et al. 2020. The eighth visual object tracking VOT2020 challenge results. In *ECCV*, 547–601. Springer.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; and Yan, J. 2019. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In *Proceedings of the CVPR*, 4282–4291. IEEE.

Li, B.; Yan, J.; Wu, W.; Zhu, Z.; and Hu, X. 2018. High Performance Visual Tracking With Siamese Region Proposal Network. In *Proceedings of the CVPR*, 8971–8980. IEEE.

Li, Y.; Mao, H.; Girshick, R.; and He, K. 2022. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the ICCV*.

Loshchilov, I.; and Hutter, F. 2018. Decoupled weight decay regularization. In *Proceedings of the ICLR*.

Mueller, M.; Smith, N.; and Ghanem, B. 2016. A benchmark and simulator for uav tracking. In *Proceedings of the ECCV*, 445–461. Springer.

Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; and Ghanem, B. 2018. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In *Proceedings of the ECCV*.

Nam, H.; and Han, B. 2016. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In *Proceedings of the CVPR*, 4293–4302. IEEE.

Pu, S.; Song, Y.; Ma, C.; Zhang, H.; and Yang, M.-H. 2018. Deep Attentive Tracking via Reciprocative Learning. In *NeurIPS*, 1931–1941.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 8821–8831. PMLR.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the CVPR*, 658–666.

Shen, Q.; Qiao, L.; Guo, J.; Li, P.; Li, X.; Li, B.; Feng, W.; Gan, W.; Wu, W.; and Ouyang, W. 2022. Unsupervised Learning of Accurate Siamese Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8101–8110.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

Song, Z.; Yu, J.; Chen, Y.-P. P.; and Yang, W. 2022. Transformer Tracking With Cyclic Shifting Window Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8791–8800.

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

Voigtlaender, P.; Luiten, J.; Torr, P. H.; and Leibe, B. 2020. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the CVPR*, 6578–6588.

Wang, G.; Luo, C.; Sun, X.; Xiong, Z.; and Zeng, W. 2020. Tracking by instance detection: A meta-learning approach. In *Proceedings of the CVPR*, 6288–6297.

Wang, N.; Zhou, W.; Wang, J.; and Li, H. 2021. Transformer Meets Tracker: Exploiting Temporal Context for Robust Visual Tracking. In *Proceedings of the CVPR*, 1571–1580.

Wei, C.; Fan, H.; Xie, S.; Wu, C.-Y.; Yuille, A.; and Feichtenhofer, C. 2022. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14668–14678.

Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; and Zhang, L. 2021. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22–31.

Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; and Hu, H. 2022. SimMIM: A Simple Framework for Masked Image Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9653–9663.

Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; and Yu, G. 2020. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the AAAI*, volume 34, 12549–12556.

Yan, B.; Peng, H.; Fu, J.; Wang, D.; and Lu, H. 2021. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the ICCV*.

Yu, Y.; Xiong, Y.; Huang, W.; and Scott, M. R. 2020. Deformable siamese attention networks for visual object tracking. In *Proceedings of the CVPR*, 6728–6737.

Zhang, Z.; Liu, Y.; Wang, X.; Li, B.; and Hu, W. 2021. Learn to match: Automatic matching network design for visual tracking. In *Proceedings of the ICCV*, 13339–13348.