Learning Target-aware Representation for Visual Tracking via Informative Interactions

Mingzhe Guo[1*], **Zhipeng Zhang**[2*], **Heng Fan**[3], **Liping Jing**[1†], **Yilin Lyu**[1], **Bing Li**[2] and **Weiming Hu**[2]
[1]Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University
[2]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[3]Department of Computer Science and Engineering, University of North Texas, Denton, TX USA
{mingzheguo, yilinlyu, lpjing}@bjtu.edu.cn, zhangzhipeng2017@ia.ac.cn,
heng.fan@unt.edu, {bli, wmhu}@nlpr.ia.ac.cn

Abstract

We introduce a novel backbone architecture to improve target-perception ability of feature representation for tracking. Having observed de facto frameworks perform feature matching simply using the backbone outputs for target localization, there is no direct feedback from the matching module to the backbone network, especially the shallow layers. Concretely, only the matching module can **directly** access the target information, while the representation learning of candidate frame is blind to the reference target. Therefore, the accumulated target-irrelevant interference in shallow stages may degrade the feature quality of deeper layers. In this paper, we approach the problem by conducting multiple branch-wise interactions **in**side the Siamese-like **b**ackbone **n**etworks (**InBN**). The core of **InBN** is a general interaction modeler (**GIM**) that injects the target information to different stages of the backbone network, leading to better target-perception of candidate feature representation with negligible computation cost. The proposed **GIM** module and **InBN** mechanism are general and applicable to different backbone types including CNN and Transformer for improvements, as evidenced on multiple benchmarks. In particular, the CNN version improves the baseline with 3.2/6.9 absolute gains of SUC on LaSOT/TNL2K. The Transformer version obtains SUC of 65.7/52.0 on LaSOT/TNL2K, which are on par with recent SOTAs.

1 Introduction

As one of the most fundamental tasks in computer vision, visual object tracking (VOT) aims to estimate the trajectory of the designated target in a video sequence [Smeulders *et al.*, 2013; Li *et al.*, 2013]. Embracing powerful deep networks for appearance modeling, unprecedented achievement has been witnessed in the past years in tracking community. One representative paradigm, namely two-stream network, has been widely adopted in recent tracking methods, *e.g.,* Siamese and Correlation-Filter based approaches [Bertinetto *et al.*, 2016; Danelljan *et al.*, 2019]. The consensus for constructing a robust two-stream tracker is to learn an informative visual representation, *i.e.*, "feature matters" [Wang and Yeung, 2013].

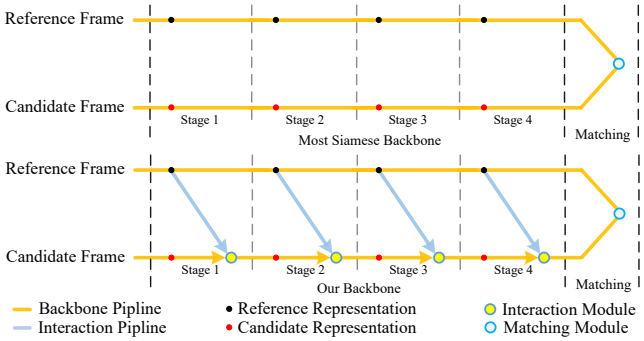

Figure 1: Comparison between two types of backbone networks in tracking. Most Siamese trackers solely use the backbone network to extract representations without interactions (the first row). In contrast, our backbone performs multiple interactions inside different stages of the backbone, which enhances the distractor-aware ability and target-aware representations learning (the second row).

For tracking task, an informative feature requires the target representation clearly distinguishable from background and objects with similar semantics (*i.e.,* distractors). Fortunately, other vision tasks, *e.g.* classification, detection and segmentation, also enjoy sharp contrast between background and objects. A new paradigm from the basic classification task, *i.e., backbone network*, can thus be seamlessly transferred to advance other fields. Object tracking also benefits from this bonus, where different backbone networks are introduced to learn strong visual representation [Zhang and Peng, 2019; Li *et al.*, 2019]. But itch of scabies for tracking mostly attributes to surrounding distractors. Besides better feature extractors, a tracking algorithm should also consider how to reinforce target-relevant representation as well as avoiding negative influence from distractors. Previous works, *e.g.,* SiamAttn [Yu *et al.*, 2020], try to alleviate this issue by injecting prior knowledge of the reference image (*i.e.,* the first frame) to enhance features of candidate frame. However, only using

*Equal Contribution.
†Corresponding author.

the target-guided candidate feature for matching without direct feedback to backbone network is not enough. The potential problem is that the branch corresponding to the candidate image cannot well "sense" information of the reference target in its feature learning. Besides, the accumulation effect of target-irrelevant interference in the shallow stages may degrade the feature quality of deeper layers. In a nutshell, the under utilization of target information during representation learning may compromise the ability of filtering unrelated distractors, which results in inferior performance.

In this paper, we set out to address the aforementioned issues by proposing a mechanism called "*interaction inside the backbone network* (**InBN**)". As discussed, the anaemic distractor-aware ability at the early learning stages blames on less exposure to prior information. Therefore, one intuitive but rational solution is to increase the branch-wise interaction frequency inside the backbone network of a two-stream paradigm. In our design, we take inspiration from recent arising vision Transformer architecture [Dosovitskiy *et al.*, 2020], which is adept on modeling global visual relations [Luo *et al.*, 2016] and can naturally process features from multi-modalities. The proposed general interaction modeler, dubbed as **GIM**, strengthens the representation of candidate image by learning its relation with reference representation. The relation learning is realized by the context-aware self-attention (CSA) and target-aware cross-attention (TCA) modules (as described later).

For the first time, we integrate the cross-processing modeler into different stages of a backbone network in visual tracking, which can continuously expose prior information to backbone modeling, as shown in Fig. 1. The **multiple interactions** pattern improves the distractor-aware ability of the early stages at a backbone network through feature aggregation in CSA and relation learning in TCA. Notably, the proposed method can not only adapt to a CNN network but also work well with a Transformer backbone. When equipping the transformer backbone with the proposed GIM and InBN mechanism, we surprisingly observe that complicated matching networks as in TransT [Chen *et al.*, 2021] and AutoMatch [Zhang *et al.*, 2021] are not necessary, where a simple cross-correlation can show promising results.

A series of experiments are conducted to prove the generality and effectiveness of the proposed LiBN and GIM. Taking SiamCAR [Guo *et al.*, 2020] as the baseline tracker, we directly apply GIM and LiBN to its CNN backbone, achieving 3.2 points gains on LaSOT (53.9 vs 50.7). When employing Transformer backbone, our model further improves the performance to SUC of 65.7 on LaSOT.

The main contributions of this work are as follows:

- We present the first work demonstrating that information flow between reference and candidate frames can be realized inside the backbone network, which is clearly different from previous feature matching way that does not provide direct feedback to backbones in visual tracking.
- We prove the effectiveness of the proposed GIM module and InBN mechanism on both CNN and Transformer backbone networks, which makes it a general modeler to improve representation qualities in tracking.

2 Related Work

Visual Object Tracking. The Siamese network has been widely used in visual object tracking in recent years [Henriques *et al.*, 2008; Bertinetto *et al.*, 2016; Li *et al.*, 2018; Li *et al.*, 2019; Zhang *et al.*, 2020; Zhang and Peng, 2019; Chen *et al.*, 2021; Danelljan *et al.*, 2019; Bhat *et al.*, 2019]. SiamFC [Bertinetto *et al.*, 2016] first adopts cross-correlation in the Siamese framework, which tracks a target by template matching. By exploiting the region proposal network (RPN) [Ren *et al.*, 2015], SiamRPN [Li *et al.*, 2018] and its variants [Guo *et al.*, 2020; Yu *et al.*, 2020] achieve more precise target scale estimation and fast speed. Later, further improvements upon Siamese trackers have been made, including enhancing feature representations by deeper convolutional backbones [Li *et al.*, 2019; Zhang and Peng, 2019], designing more practical update mechanism [Zhang *et al.*, 2019], introducing anchor-free regression to target estimation [Guo *et al.*, 2020; Zhang *et al.*, 2020], replacing cross-correlation with the automatically learned matching networks [Zhang *et al.*, 2021], introducing vision Transformer [Chen *et al.*, 2021], and so on. Another important branch, *i.e.,* online trackers, always construct models with discriminative correlation filter [Danelljan *et al.*, 2017; Danelljan *et al.*, 2019; Bhat *et al.*, 2019]. Despite of their success, only relying on a sole backbone to extract image representation without any interactive learning may introduce distractors and limit the tracking performance. In this work, we introduce a general interaction modeler to alleviate this issue.

Vision Transformer. Transformer is originally proposed in [Vaswani *et al.*, 2017] for the task of machine translation. Recently it has been introduced into vision tasks and shows great potential. The core idea of Transformer is self-attention mechanism, which takes a sequence as the input, and builds similarity scores of each two feature vectors as attention weights to reinforce or suppress visual representations of corresponding positions. ViT [Dosovitskiy *et al.*, 2020] and its follow ups [Liu *et al.*, 2021; Chu *et al.*, 2021], adopt a convolution-free transformer architecture for image classification, which processes the input image into a sequence of patch tokens. In addition to classification, Transformer is also used in other vision tasks and achieves comparable performances, such as object detection [Carion *et al.*, 2020; Sun *et al.*, 2020b], semantic segmentation [Strudel *et al.*, 2021], multiple object tracking [Sun *et al.*, 2020a], etc. In visual tracking, TrDiMP [Wang *et al.*, 2021a] uses transformer to learn temporal relations between frames, and TransT [Chen *et al.*, 2021] employs cross-attention in matching process to enhance semantic messages. Transformer is naturally suitable for processing multi-modality features. Surprisingly, no prior works in visual tracking consider building interactions inside the two-stream backbone network. The information in the reference image (or template image named in some works) is crucial to explore more target-relevant clues and track targets in complex scenes, especially in the presence of similar distractors. Our work aims to narrow these gaps by proposing an general interaction modeler which promotes the representation of candidate image at different stages of the backbone network.

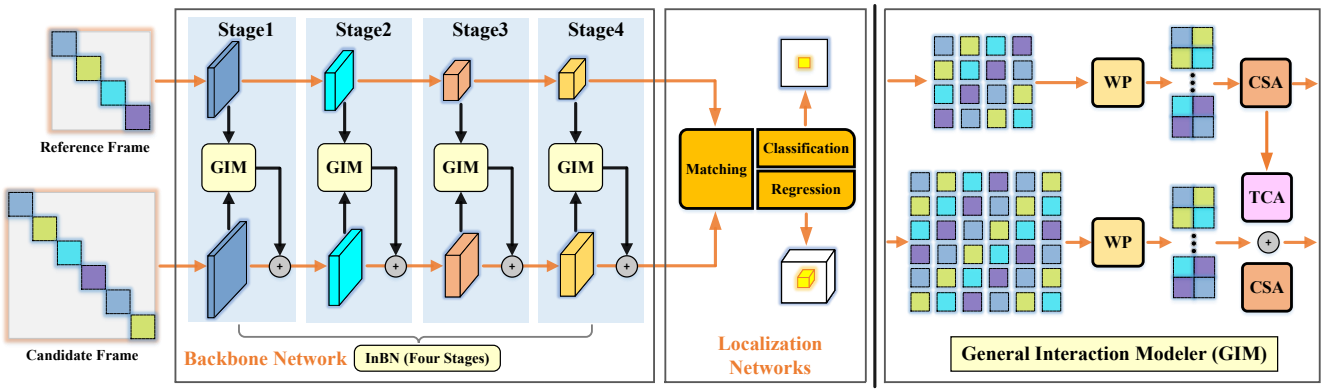

Figure 2: Architecture of the proposed tracking framework with General Interaction Modeler **(GIM)**. The window process (**WP**) ensures the successive attention modules consider both global and local context. The context-aware self-attention **(CSA)** module and target-aware cross-attention (**TCA)** module are introduced to improve feature representation learning. Notably, our method can be applied to both CNN backbones like ResNet and Transformer backbones.

3 Tracking with General Interaction Modeler

In this section, we elaborate on the proposed General Interaction Modeler (**GIM**) and its integration to different backbone networks via the proposed **InBN** mechanism for improving representation learning in visual tracking.

3.1 General Interaction Modeler

General Interaction Modeler (GIM) contains three essential components including window process (WP), context-aware self-attention (CSA) module and target-aware cross-attention (TCA) module, as illustrated in Fig. 2. The WP module is designed to decrease computational complexity and meanwhile increase the local perception ability. The CSA module aggregates context information by global attention modeling. The TCA module enhances target-related features and suppresses distractor responses by injecting prior knowledge of the reference image to representation learning. In the following, we will detail each module one by one.

Local Perception via Window Process (WP)

Our GIM is built upon the Transformer-like structure as described above. The Achilles' Heel is its quadratic cost for global attention learning and lack of local perception ability [Liu *et al.*, 2021; Chu *et al.*, 2021]. Motivated by recent vision transformers [Liu *et al.*, 2021], we design the window process to merge feature vectors of non-overlapping local areas before attention modules.

Window process (WP) first partitions the input feature $\mathbf{f} \in \mathbb{R}^{B\times H\times W\times C}$ (B is the batch size) by performing non-overlapping $win \times win$ windows ($win = 7$) on it. Then each group containing win^2 feature vectors is regarded as a modeling unit and forms a new dimension with size of $\mathbf{f}_{group} \in \mathbb{R}^{B\times win^2\times \frac{H}{win}\times \frac{W}{win}\times C}$. To fit the requirement of following attention modules, we flatten the spatial dimension of $\mathbf{f}_{group}$ to $\mathbf{f}_{WP} \in \mathbb{R}^{B\times win^2\times \frac{HW}{win^2}\times C}$, whose sequence length is $\frac{HW}{win^2}$. We perform attention on dimensions of $\frac{HW}{win^2}$ and C instead of win^2 and C as in Swin-Trans [Liu *et al.*, 2021], which ensures the feasibility to compute the cross-attention weights $\mathbf{W}_{attn} \in \mathbb{R}^{B\times win^2\times \frac{H_x W_x}{win^2}\times \frac{H_z W_z}{win^2}}$ with Einstein Summation Convention [Weisstein, 2014] (z and x represent the reference and candidate images respectively). By regarding each window as a computing unit and then modeling relations between different windows, local clues are used without affecting global receptive fields.

Context-aware Self-attention (CSA)

The proposed CSA aggregates context information by performing multi-head self-attention (MHSA) on the partitioned input feature $\mathbf{f}_{WP}$. In particular, given an input feature sequence $\mathbf{f}_{seq} \in \mathbb{R}^{B\times HW\times C}$, the inner single-head attention function first maps it into query $\mathbf{Q} \in \mathbb{R}^{B\times N_Q\times d}$, key $\mathbf{K} \in \mathbb{R}^{B\times N_K\times d}$ and value $\mathbf{V} \in \mathbb{R}^{B\times N_V\times d}$ (N_Q, N_K and N_V are sequence lengths) with mapping weights $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{C\times d}$, then performs dot-product on them,

$$\begin{aligned} \mathbf{Q} = \mathbf{f}_{seq}\mathbf{W}_Q, \mathbf{K} = \mathbf{f}_{seq}\mathbf{W}_K, \mathbf{V} = \mathbf{f}_{seq}\mathbf{W}_V, \\ \text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}, \end{aligned} \tag{1}$$

where d is the number of channels. In our model, we directly split the features along channel dimension for multiple heads, which is computed as follows,

$$\begin{aligned} \text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) &= \text{Concat}(\mathbf{H}_1,...,\mathbf{H}_n)\mathbf{W}_{map}, \\ \mathbf{H}_i &= \text{Attention}(\mathbf{Q}_i,\mathbf{K}_i,\mathbf{V}_i), \\ \mathbf{Q} &= \text{Concat}(\mathbf{Q}_1,...,\mathbf{Q}_n), \\ \mathbf{K} &= \text{Concat}(\mathbf{K}_1,...,\mathbf{K}_n), \\ \mathbf{V} &= \text{Concat}(\mathbf{V}_1,...,\mathbf{V}_n), \end{aligned} \tag{2}$$

where $\mathbf{Q}_i \in \mathbb{R}^{B\times N_Q\times \frac{d}{n}}$, $\mathbf{K}_i \in \mathbb{R}^{B\times N_K\times \frac{d}{n}}$, $\mathbf{V}_i \in \mathbb{R}^{B\times N_V\times \frac{d}{n}}$ are divided parts, and $\mathbf{W}_{map} \in \mathbb{R}^{d\times C}$ is mapping weights. We employ $n = 4, C = 256$, and $d = 256$ to achieve real-time tracking speed without other specified.

CSAs are applied on features of both reference and candidate images. It computes similarity scores between each two positions and uses the scores as attention weights to aggregate context visual features. By considering $\mathbf{f}_{WP}$ as a sequence, the sequence length after WP is $\frac{HW}{win^2}$ instead of

original HW, which has linear cost and promotes modeling speed. In a nutshell, the mechanism of CSA for the input $\mathbf{f}_{WP} \in \mathbb{R}^{B\times win^2\times \frac{HW}{win^2}\times C}$ can be summarized as

$$\begin{aligned}\mathbf{Q}=\mathbf{f}_{WP}\mathbf{W}_Q, \mathbf{K}=\mathbf{f}_{WP}\mathbf{W}_K, \mathbf{V}=\mathbf{f}_{WP}\mathbf{W}_V,\\ \mathbf{f}_{CSA}=\mathbf{f}_{WP}+\mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}),\end{aligned} \tag{3}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{C\times d}$ are mapping weights, $\mathbf{f}_{CSA} \in \mathbb{R}^{B\times win^2\times \frac{HW}{win^2}\times C}$ is the output of CSA.

Target-aware Cross-attention (TCA)

As mentioned before, Siamese tracking is conducted on a two-stream network, which is different from the typical one-stream classification model. Therefore, it is reasonable to consider the interaction between these two branches during backbone design. In this work, we propose the target-aware cross-attention (TCA) to inject the reference information to the branch of the candidate image, which guides the backbone to perceive the target and filter irrelevant distractors.

For implementation, TCA takes two window-processed features ($\mathbf{f}_{x_{WP}} \in \mathbb{R}^{B\times win^2\times \frac{H_xW_x}{win^2}\times C}$ for candidate image and $\mathbf{f}_{z_{WP}} \in \mathbb{R}^{B\times win^2\times \frac{H_zW_z}{win^2}\times C}$ for reference image) as inputs and maps them to $\mathbf{Q}$ and $\mathbf{K}, \mathbf{V}$ respectively. Through multi-head attention, $\mathbf{f}_{x_{WP}}$ are refined by the prior knowledge in $\mathbf{f}_{z_{WP}}$, where the target-related ones are strengthened and the irrelevant ones are suppressed. TCA is formulated as,

$$\begin{aligned}\mathbf{Q}=&\mathbf{f}_{x_{WP}}\mathbf{W}_Q, \mathbf{K}=\mathbf{f}_{z_{WP}}\mathbf{W}_K, \mathbf{V}=\mathbf{f}_{z_{WP}}\mathbf{W}_V,\\ &\mathbf{f}_{TCA}=\mathbf{f}_{x_{WP}}+\mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}),\end{aligned} \tag{4}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{C\times d}$ are mapping weights, $\mathbf{f}_{TCA} \in \mathbb{R}^{B\times win^2\times \frac{H_xW_x}{win^2}\times C}$ is the output of TCA.

Here we analyze how WP could make the computation of attention learning linear complexity. Without loss of generality, we assume $H\%win = 0$ and $W\%win = 0$. Taking each window as the computing unit, the sequence length is $\frac{HW}{win^2}$. CSA or TCA first computes relations between feature vectors of the same position in different windows, and the computation cost is $\mathcal{O}(\frac{H^2W^2}{win^4}d)$. Then the total cost for win^2 positions is $\mathcal{O}(\frac{H^2W^2}{win^2}d)$. If we let $k_1 = \frac{H}{win}$ and $k_2 = \frac{W}{win}$, the cost can be computed as $\mathcal{O}(k_1k_2HWd)$, which is significantly more efficient when $k_1 \ll H$ and $k_2 \ll W$ and grows linearly with HW if k_1 and k_2 are fixed.

3.2 Building Backbone Network

To demonstrate the generality and effectiveness of the proposed GIM, we follow the InBN mechanism and build two types of backbone networks for Siamese tracking, *i.e.,* CNN and Transformer as shown in Tab. 1. To unleash the target-perception ability of shallow layers, we adopt a hierarchical design to employ the GIM, which is named as InBN mechanism. We arrange the proposed GIM to different stages of a tracking backbone to conduct **multiple interactions**. Based on the reference presentation of different resolutions, the network knows what target is the one being tracked, which eventually enhances target-related messages and suppresses distractors hierarchically.

	CNN Backbone			Transformer Backbone		
	Output Size	Layer Name	Parameter	Output Size	Layer Name	Parameter
Stage 1	$\frac{H}{4}\times\frac{W}{4}$	Convolution Block	$P_1=4; C_1=256$	$\frac{H}{4}\times\frac{W}{4}$	Patch Embedding	$P_1=4; C_1=96$
		GIM	$\begin{bmatrix} WP \\ CSA \\ (TCA) \end{bmatrix}\times 1$		GIM	$\begin{bmatrix} WP \\ CSA \\ (TCA) \end{bmatrix}\times 2$
Stage 2	$\frac{H}{8}\times\frac{W}{8}$	Convolution Block	$P_2=2; C_2=512$	$\frac{H}{8}\times\frac{W}{8}$	Patch Embedding	$P_2=2; C_2=192$
		GIM	$\begin{bmatrix} WP \\ CSA \\ (TCA) \end{bmatrix}\times 1$		GIM	$\begin{bmatrix} WP \\ CSA \\ (TCA) \end{bmatrix}\times 2$
Stage 3	$\frac{H}{8}\times\frac{W}{8}$	Convolution Block	$P_3=1; C_3=1024$	$\frac{H}{16}\times\frac{W}{16}$	Patch Embedding	$P_3=2; C_3=384$
		GIM	$\begin{bmatrix} WP \\ CSA \\ (TCA) \end{bmatrix}\times 1$		GIM	$\begin{bmatrix} WP \\ CSA \\ (TCA) \end{bmatrix}\times 6$
Stage 4	$\frac{H}{8}\times\frac{W}{8}$	Convolution Block	$P_4=1; C_4=2048$	$\frac{H}{16}\times\frac{W}{16}$	Patch Embedding	$P_4=1; C_4=768$
		GIM	$\begin{bmatrix} WP \\ CSA \\ (TCA) \end{bmatrix}\times 1$		GIM	$\begin{bmatrix} WP \\ CSA \\ (TCA) \end{bmatrix}\times 2$
Output	$\frac{H}{8}\times\frac{W}{8}$	Mapping	$C=256$	$\frac{H}{8}\times\frac{W}{8}$	Fusion	$C=256$

Table 1: Configurations of the built backbone with GIM and InBN.

For the CNN-based model, we take the ResNet-50 in SiamCAR [Guo *et al.*, 2020] as the basic CNN backbone. Following the InBN paradigm, we modify it by directly applying the GIM after each stage. The mapping layers are attached at each stage to unify the output feature dimensions to 256 (see more details in [Guo *et al.*, 2020]).

For the Transformer-based model, we refer to the structure of Swin-Trans [Liu *et al.*, 2021] to build two-stream backbone by stacking the GIMs. As illustrated in Tab. 1, the backbone network is divided into four stages, which contain 2, 2, 6, 2 GIMs, respectively. Notably, we only use the TCA module in the last GIM of each stage, which avoids bringing high computation costs. The InBN mechanism allows backbone to enhance target-relevant ability gradually by hierarchical interactions. For the output features of each stage, we first map their channel dimensions into the same number of 256 with a 1×1 linear layer. We upsample or downsample these features to same resolution to align spatial dimension. Finally, we concatenate all features along channel dimension and use a 1×1 linear transformation to shrink channel number to 256.

3.3 Comparison with Other Tracking Backbones

To our best knowledge, this is the first tracking framework that considers branch-wise interactions inside the backbone network for better feature representation. As shown in Fig. 1, we categorize existing backbone networks in visual tracking into two types. The first one (see the first row in Fig. 1), which is employed in the most trackers, simply uses a backbone to extract feature representation and then performs matching without direct feedback for tracking. The lack of reference information makes it hard to robustly locate the target object. Differently, equipping with GIM with InBN mechanism, our backbone network (see the second row in Fig. 1) is able to not only perform multiple interactions to bridge the target information to candidate frame during feature learning, but also exploit global modeling to enrich target-related representation, resulting in improved feature representations in all layers for better performance, yet with negligible computation.

Method	Source	LaSOT			TNL2K			UAV123		NFS		FPS
		AUC	P_{Norm}	P	AUC	P_{Norm}	P	AUC	P	AUC	P	
TransInMo*	**Ours**	**65.7**	**76.0**	**70.7**	**52.0**	**58.5**	**52.7**	**69.0**	**89.0**	**66.8**	**80.2**	34
TransInMo	**Ours**	**65.3**	**74.6**	**69.9**	**51.5**	**57.8**	**52.6**	**68.2**	**90.5**	**66.4**	**80.7**	**67**
TransT	CVPR2021	**64.9**	**73.8**	**69.0**	**50.7**	**57.1**	**51.7**	**68.1**	**87.6**	65.7	**78.8**	50
TrDiMP	CVPR2021	63.9	-	61.4	-	-	-	67.5	87.2	**66.5**	78.4	26
CNNInMo	**Ours**	53.9	61.6	53.9	42.2	48.6	41.9	62.9	81.8	56.0	68.5	47
SiamCAR	CVPR2020	50.7	60.0	51.0	35.3	43.6	38.4	61.4	76.0	52.9	65.7	52
AutoMatch	ICCV2021	58.3	-	59.9	47.2	-	43.5	63.9	-	60.6	-	50
Ocean	ECCV2020	56.0	65.1	56.6	38.4	45.4	37.7	62.1	-	55.3	-	58
KYS	ECCV2020	55.4	63.3	-	44.9	51.1	43.5	-	-	63.5	77.0	20
SiamFC++	AAAI2020	54.4	62.3	54.7	38.6	45.0	36.9	63.1	76.9	58.1	-	90
PrDiMP	CVPR2020	59.8	68.8	60.8	47.0	54.0	45.9	66.8	87.6	63.5	76.5	30
CGACD	CVPR2020	51.8	62.6	53.5	-	-	-	63.3	83.3	-	-	70
SiamAttn	CVPR2020	56.0	64.8	-	-	-	-	65.0	84.5	-	-	45
SiamBAN	CVPR2020	51.4	59.8	52.1	41.0	48.5	41.7	63.1	83.3	59.4	70.0	40
DiMP	ICCV2019	56.9	65.0	56.7	44.7	51.3	43.4	65.4	85.6	62.7	75.1	40
SiamPRN++	CVPR2019	49.6	56.9	49.1	41.3	48.2	41.2	61.3	80.7	50.8	50.9	35
ATOM	CVPR2019	51.5	57.6	50.5	40.1	46.5	39.2	65.0	85.8	59.0	69.4	30
SiamPRN	CVPR2018	49.6	56.9	49.1	32.9	36.2	28.1	55.7	71.0	48.8	56.7	160
ECO	ICCV2017	32.4	33.8	30.1	32.6	37.7	31.7	52.5	68.8	46.6	54.7	60
SiamFC	ECCV2016	33.6	42.0	33.9	29.5	45.0	28.6	48.5	64.8	-	-	58

Table 2: Comparisons on LaSOT, TNL2K, UAV123 and NFS. The best three results are shown in **red**, **green** and **blue** fonts, respectively.

4 Experiments

4.1 Implementation Details

For CNN-based tracker, we apply the proposed **In**teraction **Mo**deler to the baseline tracker SiamCAR [Guo *et al.*, 2020] (named as **CNNInMo**). For transformer-based tracker (named as **TransInMo**), we build backbone network following the structions in Sec. 3.2. A simple depth-wise correlation layer is used for feature matching. Two three-layer MLPs are respectively conducted for classification and regression heads, as similar in [Chen *et al.*, 2021]. Moreover, we test the influence of complicated matching process by equipping TransInMo with the matching module from TransT [Chen *et al.*, 2021] (named as **TransInMo***). The training settings for CNNInMo/TransInMo follow the baseline trackers SiamCAR and TransT, respectively (we recommend the readers to [Guo *et al.*, 2020] and [Chen *et al.*, 2021] for more details). In particular, the training splits of LaSOT [Fan *et al.*, 2019], TrackingNet [Muller *et al.*, 2018] and GOT-10k [Huang *et al.*, 2019] and COCO [Lin *et al.*, 2014] are used during learning.

4.2 State-of-the-art Comparisons

We compare our methods with recent state-of-the-art trackers on six tracking benchmarks. The detailed comparison results on the LaSOT [Fan *et al.*, 2019], TNL2K [Wang *et al.*, 2021b], TrackingNet [Muller *et al.*, 2018], UAV123 [Mueller *et al.*, 2016], NFS [Kiani Galoogahi *et al.*, 2017] and OTB100 [Wu *et al.*, 2015] are reported in Tab. 2 and Tab. 3. Notably, CNNInMo runs at **47** *fps* on GTX2080Ti GPU, while TransInMo/TransInMo* run at **67/34** *fps* respectively.

LaSOT is a large-scale tracking benchmark containing 1,400 videos. We evaluate different tracking algorithms on its 280 testing videos. As reported in Tab. 2, when applying GIM and InBN to SiamCAR without any other modifications, CNNInMo obtains gains of 3.2/2.9 points on success (SUC) and precision (P) scores, respectively. This evidences the effectiveness of the proposed interaction module and mechanism. Our transformer version with simple cross-correlation matching outperforms recent state of the arts TransT [Chen *et al.*, 2021] and TrDiMP [Wang *et al.*, 2021a].

	TransInMo*	TransInMo	CNNInMo	TransT	SiamCAR
TrackingNet	81.7	81.6	72.1	81.4	65.3
OTB100	71.1	70.6	70.3	69.4	70.0

Table 3: Results comparison on TrackingNet and OTB100 (AUC).

TNL2K is a recently proposed large-scale benchmark for tracking by natural language and bounding box initialization. We evaluate our tracker on its 700 testing videos. CNNInMo outperforms the baseline tracker SiamCAR for 6.9/3.5 points on SUC and precision, respectively.

UAV123 contains 123 aerial sequences captured from a UAV platform. As shown in Tab. 2, the proposed methods (Transformer versions) perform the best among all compared trackers. The CNN version CNNInMo exceeds 1.5 SUC points than the baseline tracker SiamCAR.

NFS consists of 100 challenging videos with fast-moving objects. We evaluate the proposed trackers on the 30 fps version, as presented in Tab. 2. TransInMo and TransInMo* obtain the best two performances on success and precision scores, which shows that our method can improve the robustness of tracking fast-tracking moving targets.

# NUM	Method	WP	TCA	LaSOT			FPS
				AUC	P_{Norm}	P	
①	TransInMo*			61.1	70.3	64.9	30
②	TransInMo*	√		64.6	74.7	68.8	37
③	TransInMo*		√	63.2	73.6	68.0	27
④	TransInMo*	√	√	**65.7**	**76.0**	**70.7**	34

Table 4: Component-wise analysis of GIM on TransInMo*.

# NUM	Method	DW	Matching in TransT		LaSOT			FPS
			Encoder Layer Num	Decoder Layer Num	AUC	P_{Norm}	P	
①	TransInMo	√			65.3	74.6	69.9	67
②	TransInMo*		0	1	65.0	74.3	69.8	59
③	TransInMo*		1	1	65.3	73.7	69.0	52
④	TransInMo*		2	1	65.5	73.7	69.0	46
⑤	TransInMo*		3	1	65.7	74.1	69.2	40
⑥	TransInMo*		4	1	**65.7**	**76.0**	**70.7**	34

Table 5: Ablation study on different matching settings.

TrackingNet and OTB100 We further evaluate proposed trackers on TrackingNet and OTB100. As presented in Tab. 3, CNNInMo surpasses the baseline SiamCAR for 6.8 points on TrackingNet. TransInMo and TransInMo* are on par with (slightly better than) TransT on these two benchmarks.

4.3 Ablation Study

Components in GIM. As shown in Tab. 4, we analyze the influence of WP and TCA in **GIM** based on **TransInMo***. CSA is the basic modeling layer in GIM, hence we only ablate the other two modules. Tab. 4 shows that without WP and TCA, the tracker obtains an AUC score of 61.1 on LaSOT. When integrating WP into the model, it brings 3.5 points gains (② *vs* ①) with faster running speed (37 *fps vs* 30 *fps*). The TCA module builds branch-wise interaction between the reference and candidate images inside the backbone. As in Tab. 4, TCA brings gains of 2.1 points on AUC (③). When applying WP and TCA together, further improvement is obtained with an AUC score of 65.7 (④), which proves the rationality of our design.

Matching Process. We ablate different matching process for our trackers as shown in Tab. 5. With simple depth-wise correlation (DW) (①, *i.e.* TransInMo), the tracker has already achieved compelling performance and speed. We then replace DW with more complex matching modules in TransT with different Transformer encoder numbers. Surprisingly, it (⑥) does not obtain significant gains. As we decrease the encoder layers, the performance is even worse than simple DW (①). Our framework indicates that a tracker with InBN mechanism reduces the demand of fussy matching network to filter target-irrelevant distractors. Matching inside the backbone may be a worth choice in future work.

Activation Map Visualization. To better understand our CSA and TCA, we visualize their activation maps in Fig. 3. As shown, when the target appearance changes heavily or similar distractors exist, the branch-wise interaction of TCA can help the backbone sense the target-relevant areas. Then self-attention of CSA employs the context messages to filter useless messages and focus on the target area.

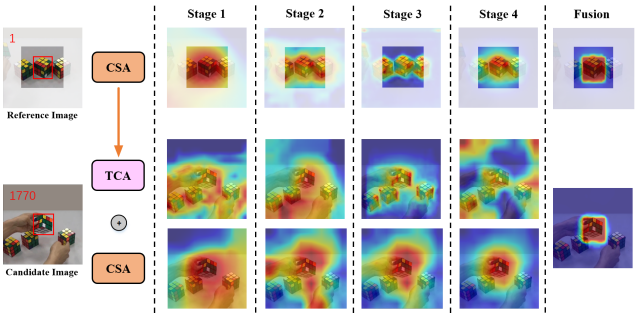

Figure 3: Activation maps for CSA and TCA.

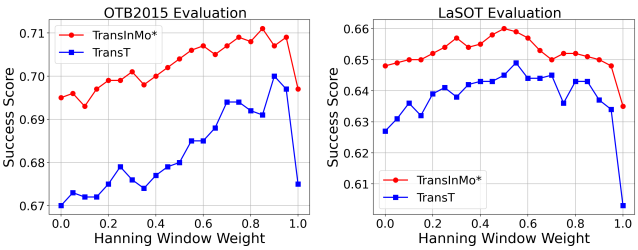

Figure 4: The influence of the Hanning window.

Hanning Window Influence. In Fig. 4, we present the influence of post-processing (i.e. Hanning window) used in our TransInMo* and TransT on two benchmarks. From Fig. 4, we observe that the proposed TransInMo* is more stable. In the case without post-processing (*i.e.* weight=0), our tracker still shows compelling performance, which indicates the effectiveness of feature representation learned by our backbone.

Due to limited space, more experiments with in-depth analysis, *e.g.*, tracking result visualization, failure case analysis, are presented in the appendix.

5 Conclusions

This work presents a novel mechanism that conducts branch-wise interactions inside the visual tracking backbone network (InBN) via the proposed general interaction modeler (GIM). We prove that both the CNN network and the Transformer backbone can enjoy bonus brought by InBN. The backbone network can build more robust feature representation under complex environments with the improved target-perception ability. Our method achieves compelling tracking performance by applying the backbones to Siamese tracking.

Acknowledgments

This work was supported by the National Key Research and Development Program (2020AAA0106800), the Beijing Natural Science Foundation under Grant (Z180006, L211016), the National Natural Science Foundation of China under Grant (62176020) and CAAI-Huawei MindSpore Open Fund. This work is co-supervised by Prof. Liping Jing and Dr. Zhipeng Zhang.

References

[Bertinetto *et al.*, 2016] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H S Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision Workshops*, 2016.

[Bhat *et al.*, 2019] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

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

[Chen *et al.*, 2021] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[Chu *et al.*, 2021] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting spatial attention design in vision transformers. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021.

[Danelljan *et al.*, 2017] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[Danelljan *et al.*, 2019] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Fan *et al.*, 2019] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[Guo *et al.*, 2020] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[Henriques *et al.*, 2008] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. In *ICVS*, 2008.

[Huang *et al.*, 2019] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 2019.

[Kiani Galoogahi *et al.*, 2017] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[Li *et al.*, 2013] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. *ACM transactions on Intelligent Systems and Technology (TIST)*, 2013.

[Li *et al.*, 2018] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[Li *et al.*, 2019] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[Luo *et al.*, 2016] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.

[Mueller *et al.*, 2016] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *European Conference on Computer Vision*, 2016.

[Muller *et al.*, 2018] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *European Conference on Computer Vision*, 2018.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R–CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems*, 2015.

[Smeulders *et al.*, 2013] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan,

and Mubarak Shah. Visual tracking: An experimental survey. *TPAMI*, 2013.

[Strudel *et al.*, 2021] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*, 2021.

[Sun *et al.*, 2020a] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020.

[Sun *et al.*, 2020b] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. *arXiv preprint arXiv:2011.10881*, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31th International Conference on Neural Information Processing Systems*, 2017.

[Wang and Yeung, 2013] Naiyan Wang and Dit Yan Yeung. Learning a deep compact image representation for visual tracking. *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2013.

[Wang *et al.*, 2021a] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[Wang *et al.*, 2021b] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[Weisstein, 2014] Eric W Weisstein. Einstein summation. *https://mathworld. wolfram. com/*, 2014.

[Wu *et al.*, 2015] Yi Wu, Jongwoo Lim, and Ming Hsuan Yang. Object tracking benchmark. *TPAMI*, 2015.

[Yu *et al.*, 2020] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[Zhang and Peng, 2019] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[Zhang *et al.*, 2019] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[Zhang *et al.*, 2020] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *European Conference on Computer Vision*, 2020.

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