





Improving the Accuracy of Tracker by Linearized Transformer

Thang Hoang Dinh¹^a, Kien Thai Trung¹^b, Thanh Nguyen Chi¹^c and Long Tran Quoc^{2,*}^d

¹*Institute of Information Technology, Academy of Military Science and Technology, Hanoi, Vietnam*

²*University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam*

Keywords: Visual Tracking, Siamese Visual Tracking, Linearized Transformer, Attention.

Abstract: Visual object tracking seeks to correctly estimate the target's bounding box, which is difficult due to occlusion, illumination variation, background clutters, and camera motion. Recently, Siamese-based approaches have demonstrated promising visual tracking capability. However, most modern Siamese-based methods compute target and search image features independently, then use correlation to acquire correlation information from two feature maps. The correlation operation is a straightforward fusion technique that considers the similarity between the template and the search region. This may be the limiting factor in the development of high-precision tracking algorithms. This research offers a Siamese refinement network for visual tracking that enhances and fuses template and search patch information directly without needing a correlation operation. This approach can boost any tracker performance and produces boxes without any postprocessing. Extensive experiments on visual tracking benchmarks such as VOT2018, UAV123, OTB100, and LaSOT with DiMP50 base tracker demonstrate that our method achieves state-of-the-art results. For example, on the VOT2018, LaSOT, and UAV123 test sets, our method obtains a significant improvement of 5.3% (EAO), 3.5% (AUC), and 2.9% (AUC) over the base tracker. Our network runs at approximately 30 FPS on GPU RTX 3070.

1 INTRODUCTION


Visual tracking is crucial in computer vision since it lets us determine the status of an item inside a video sequence. Despite significant gains in recent years, illumination variations, background clutters, occlusion, and camera motion impede visual tracking. Numerous research has been published in recent years, but it is still necessary to produce an accurate method.


The Siamese network-based tracker (Li et al., 2019; Chen et al., 2020; Yu et al., 2020) formulate the problem of visual object tracking as learning a generic similarity map by cross-correlating the feature representations of the template target and search area. However, cross-correlation is a linear matching procedure, limiting the tracker's capacity to capture the complex non-linear interaction between the template and search patch. In addition, these Siamese-based trackers often identify the target by separately improving the regression and classification branches, which might result in a tracking technique mismatch.


DiMP (Bhat et al., 2019) and KYS (Bhat et al., 2020) use a multiple-stage tracking technique, which incorporates extra tracking stages for more precise box estimates, to provide more robust and accurate tracking outcomes. These trackers begin by approximating the target's location before refining the original result in subsequent tracking phases for a more exact box prediction.


Recently, the attention and transformer mechanism was introduced to visual tracking in (Yu et al., 2020; Wang et al., 2021; Zhao et al., 2021; Chen et al., 2021). SiamAttn (Yu et al., 2020) is an anchor-based tracker that analyzes both self- and cross-attention to enhance the discriminative ability of the template and search features before performing depth-wise cross-correlation. TransformerTrack (Wang et al., 2021) utilizes a whole transformer to provide a tracking framework with transformer assistance. TrTr (Zhao et al., 2021) propose a tracker network based on a powerful attention mechanism called Transformer encoder-decoder architecture.

In addition, the majority of refinement techniques in current trackers (Bhat et al., 2019; Bhat et al., 2020; Cheng et al., 2021) have poor transferability since their training is connected with other components. And Alpha-Refine (Yan et al., 2021) continues to use correlation. Nevertheless, a correlation technique can

^a <https://orcid.org/0000-0002-6099-7522>

^b <https://orcid.org/0000-0002-3098-814X>

^c <https://orcid.org/0000-0003-4335-7002>

^d <https://orcid.org/0000-0002-4115-2890>

*Corresponding author

only determine the link between small patches in two feature maps. Similar Alpha-Refine and in contrast to the abovementioned methodologies, our methodology is independently trained. Therefore, it can be directly applied to any existing trackers in a plug-and-play style without further training or change of the baseline tracker.

In this study, we propose and develop a unique Feature Enhancement module (FEM) for the enhancement and a Feature Fusion module (FFM) for the fusion of two Siamese branch features, therefore eliminating the aforementioned issue. The FEM module repeatedly interweaves the self and cross-encoder layers, whereas FFM is an attention-based pixel-wise match. Moreover, as a consequence of predicting box coordinates using a corner heatmap. These are our most important contributions:

- We propose a new architecture that integrates feature extraction, features enhancement and fusion (FEF), and prediction head modules to improve tracker accuracy.
- The proposed FEF enriches and aggregates extensive contextual information between the template target and the search image. In addition, a linear transformer is utilized to lower the computing complexity of our framework.
- We conduct extensive experiments on multiple benchmark datasets, including VOT2018, UAV123, OTB100, and LaSOT with base tracker DiMP, demonstrating that our network achieves a good trade-off between efficiency and precision. On the VOT2018, LaSOT, and UAV123 test sets, our method obtains a significant improvement of 5.3% (EAO), 3.5% (AUC), and 2.9% (AUC) over the base tracker. Our network runs at 30 FPS on NVIDIA GeForce RTX 3070.

2 RELATED WORK

Due to the emergence of new benchmark datasets, visual tracking has been an important area of study in computer vision for the last several decades. This section provides a concise overview of the three factors most pertinent to our work.

Visual Object Tracking. Deep learning has successfully permeated computer vision for a variety of applications, including object tracking. Several trackers based on deep learning train an online classifier to differentiate the target from the backdrop and detractors. The DiMP (Bhat et al., 2019) tracker improves the discriminative capabilities of the learned CNN kernel in an end-to-end manner. Moreover, the newly-

introduced KYS (Bhat et al., 2020) extends DiMP by using scene information to enhance the outcomes.

Recently, Siamese-based trackers (Li et al., 2019; Zhang et al., 2020; Chen et al., 2020; Guo et al., 2020; Cheng et al., 2021) have garnered significant attention for their exceptional performance. SiamRPN++ (Li et al., 2019) incorporates contemporary deep networks into Siamese trackers, such as ResNet ResNet (He et al., 2016). Moreover, SiamBAN (Chen et al., 2020) and SiamCAR (Guo et al., 2020) used the FCOS (Tian et al., 2019) idea for tracking and developed a basic yet effective anchor-free tracker. These works still depend significantly on the correlation operation fusion of template and search region features. SiamRN (Cheng et al., 2021) presents a Relation Detector (RD) equipped with a contrastive training approach that is meta-trained to acquire the capacity to learn to filter the distractors from the target area by quantifying their connections. In addition, SiamGAT (Guo et al., 2021) demonstrated a target-aware Siamese Graph Attention network for generic object tracking.

Attention and Transformer Mechanism. The transformer (Vaswani et al., 2017) receives a series as input, examines each element in the sequence, and discovers their relationships. This characteristic makes the transformer capable of collecting global information in sequential data. Attention and Transformer mechanisms have also been investigated lately in object tracking (Yu et al., 2020; Wang et al., 2021; Chen et al., 2021; Zhao et al., 2021). SiamAttn (Yu et al., 2020) explores self-attention and cross-attention to improve the discriminative power of target features and then fuses features derived from the template and search images using depth-wise cross-correlation. TransformerTrack (Wang et al., 2021) employed a complete transformer consisting of an encoder and decoder that was computationally intensive, memory intensive, and slow to train. TransT (Chen et al., 2021) created a transformer-based fusion network for the inclusion of target-search data. However, box generation was still dependent on postprocessing using these approaches.

Due to their quadratic complexity concerning the input length, transformers are unacceptably slow when processing very lengthy sequences. Recent research has suggested “linear Transformers” with the memory of constant size and time complexity proportional to sequence length (Schlag et al., 2021). This decrease in complexity is mostly due to the linearization of the softmax.

Refinement Mechanism. Numerous state-of-the-art trackers (Bhat et al., 2019; Bhat et al., 2020; Cheng et al., 2021) use a multi-stage tracking method to get precise and reliable results. This strategy begins with

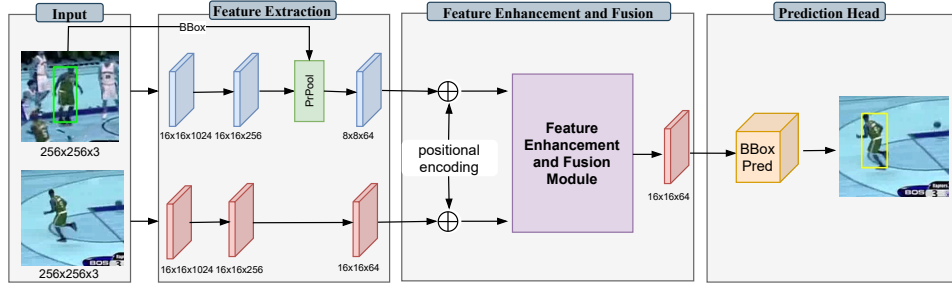


Figure 1: A summary of the proposed networks. It includes an input, feature extraction, feature enhancement and fusion, and a regression prediction head.

a coarse target location and then uses a refinement module to refine the findings of the previous stage. DiMP (Bhat et al., 2019) first locates the target using an online classification module and then draws random samples around it. Then, to offer more accurate bounding boxes, they optimize the overlap between these samples and the ground truth using a modified IoU-Net (Jiang et al., 2018). This updated IoU-Net can be trained independently of the original tracker. Consequently, the IoU-Net has excellent transferability, although its accuracy may still be significantly enhanced. SiamRN (Cheng et al., 2021) designed a refinement module that can perform classification and regression concurrently to localize the target, hence minimizing the mismatch between the two branches. However, SiamRN is meant as a standalone tracker and not as a refinement module; therefore, it cannot and should not be used to improve other trackers. Alpha-Refine (Yan et al., 2021) is a plug-and-play refinement module that improves the tracking performance of many kinds of trackers. Alpha-Refine continues to employ correlation. The correlation operation is a straightforward fusion technique that takes into account the similarity between the template and the search area. However, the correlation operation itself is a local linear matching process, resulting in the loss of semantic information and the easy occurrence of local optimum, which may be the bottleneck in the design of accurate tracking algorithms.

In this research, we use the core principles of linearized transformer and attention to constructing a Siamese refinement network for visual tracking that enhances and fuses template and search patch information directly without needing a correlation operation. In addition, as a result of employing a corner heatmap to estimate box location, that is anchor-free.

3 PROPOSED METHOD

We describe the details of our proposed networks (TrackerLT) in this section. As shown in Figure 1,

TrackerLT consists of three main components: a feature extraction, a feature enhancement and fusion, and a prediction head network.

3.1 Feature Extraction

In this work, we use the fully convolutional network to construct the Siamese subnetwork for the visual feature extraction. The Siamese network consists of two identical branches. For feature extraction, we employ a ResNet50 (He et al., 2016) pre-train on (Russakovsky et al., 2015) as the backbone network. We only use the fourth stage’s (layer3) outputs as final outputs. The backbone processes the template patch $z \in \mathbb{R}^{3 \times H_0 \times W_0}$ and the search patch $x \in \mathbb{R}^{3 \times H_0 \times W_0}$ to obtain their features maps $F_z \in \mathbb{R}^{C_z \times H \times W}$ and $F_x \in \mathbb{R}^{C_x \times H \times W}$, $H = \frac{H_0}{16}$, $W = \frac{W_0}{16}$ and $C_z = C_x = 1024$. Then, we apply a neck with three stacked convolution 1×1 , batch norm, relu to decrease the output features channel to $C=64$. The output features of our network are defined as $Z \in \mathbb{R}^{C \times H \times W}$ and $X \in \mathbb{R}^{C \times H \times W}$.

With b is the bounding box of the target object in template patch, we convert b to RoI format to get r . Then apply RoI Pooling to Z ; we get RoI feature:

$$\mathbf{Z} = \psi(Z, r) \in \mathbb{R}^{C \times h \times w} \quad (1)$$

where ψ is Precise RoI Pooling (Jiang et al., 2018).

3.2 Feature Enhancement Module

Transformer (Vaswani et al., 2017) adopts attention mechanism with Query-Key-Value (QKV) model. Given the packed matrix representations of queries $Q \in \mathbb{R}^{N \times D_k}$, keys $K \in \mathbb{R}^{M \times D_v}$, and values $V \in \mathbb{R}^{M \times D_v}$, the scaled dot-product attention used by Transformer is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right)V = AV. \quad (2)$$

where N and M denote the lengths of queries and keys (or values); D_k and D_v denote the dimensions

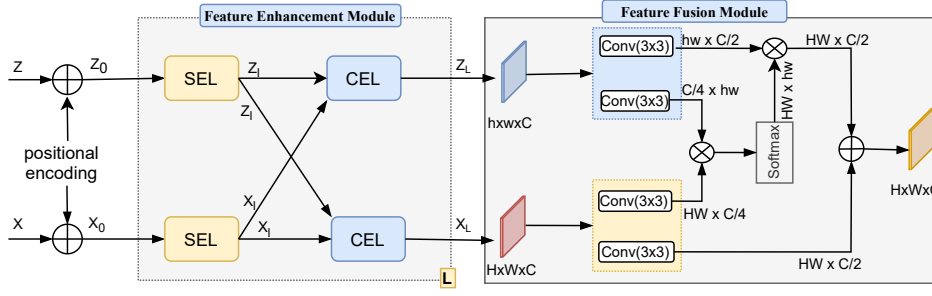


Figure 2: The proposed Feature Enhancement module (left) consists of a stack of several L Self-Encoder Layer (SEL) sub-module and several L Cross-Encoder Layer (CEL) sub-module. The proposed Feature Fusion module (right) is a pixel-wise match based on attention.

of keys (or queries) and values; $A = \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right)$ is often called an attention matrix. For softmax attention, the complexity of computing $\text{softmax}(QK^T)V$ is quadratic $O(N^2)$. Following (Schlag et al., 2021), by replacing the unnormalized attention $\exp(QK^T)$ with $\phi(Q) \cdot \phi(K)^T$ the computational complexity of attention can be reduced to $O(N)$, where ϕ is a feature map that is applied in a row-wise manner. Specifically, given an input $x \in \mathbb{R}^D$, the feature map $\phi: \mathbb{R}^D \rightarrow \mathbb{R}^{2vD}$ is defined by the partial function:

$$\phi_{i+2(j-1)D}(x) = \text{ReLU}([x, -x])_i \text{ReLU}([x, -x])_{i+1} \quad \text{for } i = 1, \dots, 2D, j = 1, \dots, v. \quad (3)$$

Hence the computation of the unnormalized attention matrix can be linearized by computing:

$$\text{Attention}(Q, K, V) = \phi(Q)(\phi(K)^T V). \quad (4)$$

As illustrated in Figure 2 (left), the proposed Feature Enhancement Module (FEM) takes Z and X as inputs, and outputs the feature enhancement by applying the linearized transformer mechanism. The FEM consists of a stacked L self-encoder layer (SEL) and cross-encoder layer (CEL).

Following (Vaswani et al., 2017), we use 2D extension function to generate spatial position encoding for input sequences Z and X :

$$Z_0 = \sigma_1(Z) + P_z \in \mathbb{R}^{C \times N_z} \quad (5)$$

$$X_0 = \sigma_2(X) + P_x \in \mathbb{R}^{C \times N_x} \quad (6)$$

where σ_1, σ_2 are two tensors reshape operators, P_z, P_x are the spatial position encodings corresponding to Z and X , respectively, $N_z = h \times w$ and $N_x = H \times W$

For SEL, suppose the input features are Z_{l-1} and X_{l-1} , $l = 1, \dots, L$, self-attention (SA) is formulated as:

$$\text{SA}(Z_{l-1}) = \phi(Z_{l-1} \mathbf{W}_Q)(\phi(Z_{l-1} \mathbf{W}_K)(Z_{l-1} \mathbf{W}_V)) \quad (7)$$

$$\text{SA}(X_{l-1}) = \phi(X_{l-1} \mathbf{W}_Q)(\phi(X_{l-1} \mathbf{W}_K)(X_{l-1} \mathbf{W}_V)) \quad (8)$$

Then, we can generate SEL features map:

$$Z_l = Z_{l-1} + \text{MPL}(\text{CAT}(Z_{l-1}, \text{SA}(Z_{l-1}))) \quad (9)$$

$$X_l = X_{l-1} + \text{MPL}(\text{CAT}(X_{l-1}, \text{SA}(X_{l-1}))) \quad (10)$$

For CEL, suppose the input features are Z_l and X_l , cross-attention (CA) is formulated as:

$$\text{CA}(Z_l) = \phi(Z_l \mathbf{W}_Q)(\phi(X_l \mathbf{W}_K)(X_l \mathbf{W}_V)) \quad (11)$$

$$\text{CA}(X_l) = \phi(X_l \mathbf{W}_Q)(\phi(Z_l \mathbf{W}_K)(Z_l \mathbf{W}_V)) \quad (12)$$

Then, we can generate CEL features map:

$$Z_l = Z_l + \text{MPL}(\text{CAT}(Z_l, \text{CA}(Z_l))) \in \mathbb{R}^{C \times N_z} \quad (13)$$

$$X_l = X_l + \text{MPL}(\text{CAT}(X_l, \text{CA}(X_l))) \in \mathbb{R}^{C \times N_x} \quad (14)$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are the learnable parameters of three linear projection layers; MPL and CAT are Multilayer Perceptron block and Concat, respectively. The output of FEM are Z_L and X_L .

3.3 Feature Fusion Module

When appearance changes or occlusions occur, detailed local features are dominant for matching the target template and search patch. Hence, instead of only using correlation operation, we propose an attention fusion mechanism where template and search features are matched at a pixel-wise level, as shown in Figure 2 (right). Key and value maps are generated from features, which serve as a means of encoding visual semantics for matching and detailed appearing information for prediction. Given Z_L and X_L from FEM, generate key and value features map by:

$$\begin{aligned} V_Z &= \sigma_1(\mathbf{W}_1(Z_L)) \in \mathbb{R}^{hw \times C/2} \\ K_Z &= \sigma_2(\mathbf{W}_2(Z_L)) \in \mathbb{R}^{C/4 \times hw} \\ K_X &= \sigma_3(\mathbf{W}_3(X_L)) \in \mathbb{R}^{HW \times C/4} \\ V_X &= \sigma_4(\mathbf{W}_4(X_L)) \in \mathbb{R}^{HW \times C/2} \end{aligned} \quad (15)$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$, and \mathbf{W}_4 are 3×3 convolution layer, respectively, $\sigma_1, \sigma_2, \sigma_3$ and σ_4 are four tensors

reshape operators. Then we calculate the similarities between key maps of template feature and search feature by:

$$A = K_Z \times K_X \quad (16)$$

where “ \times ” is the matrix dot-product operation. Then, we perform softmax normalization as:

$$S = \text{softmax}(A) \in \mathbb{R}^{HW \times hw} \quad (17)$$

Then calculate the embedding value and concat this value with the value map of the search feature to generate attention fusion feature as:

$$E = S \times V_Z \in \mathbb{R}^{HW \times C/2} \quad (18)$$

$$F = \text{concat}(V_X, E) \in \mathbb{R}^{H \times W \times C} \quad (19)$$

F contains massive information for prediction box.

3.4 Prediction Head Network

To improve the box estimation quality, we design a prediction head through estimating the probability distribution of the box corners. The fusion feature F fed into a simple fully-convolutional network (FCN). The FCN consists of four stacked Conv-BN-ReLU layers followed by a Conv layer predicting two heatmaps, which represent the top-left corner and bottom-right corner respectively. We apply softmax (Luvizon et al., 2019) to the heatmaps to outputs two probability maps $P_{tl}(x, y)$ and $P_{br}(x, y)$ for the top-left and the bottom-right corners of object bounding boxes, respectively. Finally, the predicted box coordinates $(\hat{x}_{tl}, \hat{y}_{tl})$ and $(\hat{x}_{br}, \hat{y}_{br})$ are obtained by computing the expectation of corners’ probability distribution as shown in Eq. (20).

$$\begin{aligned} (\hat{x}_{tl}, \hat{y}_{tl}) &= \left(\sum_{y=0}^H \sum_{x=0}^W x \cdot P_{tl}(x, y), \sum_{y=0}^H \sum_{x=0}^W y \cdot P_{tl}(x, y) \right), \\ (\hat{x}_{br}, \hat{y}_{br}) &= \left(\sum_{y=0}^H \sum_{x=0}^W x \cdot P_{br}(x, y), \sum_{y=0}^H \sum_{x=0}^W y \cdot P_{br}(x, y) \right) \end{aligned} \quad (20)$$

3.5 Loss Function

The box localization losses are calculated using the IoU loss and are defined as follows:

$$L_{box} = 1 - \frac{1}{N_{pos}} \sum_{i,j} 1^{obj} L_{IoU}(p_{i,j}, g_{i,j}) \quad (21)$$

where N_{pos} denotes the number of positive samples, 1^{obj} is the indicator function for positive samples, L_{IoU} denotes the IoU loss as UnitBox (Yu et al., 2016), $g_{i,j}$ denotes the ground-truth box, $p_{i,j}$ denotes the prediction bounding box.

3.6 Tracking Phase

Bhat et al. proposed DiMP (Bhat et al., 2019), which can predict the bounding box of the object in benchmarks datasets without finding hyperparameters cosine windows, penalty, and learning rate as Siamese-based method (such as SiamRPN++, SiamCAR, SiamBAN, SiamAttn, SiamGAT). Based on Alpha-Refine, we crop the initial frame’s template patch and provide it into the base tracker (DiMP) and TrackerLT during tracking. For the following frames, we trim the search patch p and pass via the base tracker to get prediction bounding-box b , then p and b process by TrackerLT again to obtain the bounding-box regression map $P_{1 \times 4}^{box} = [\hat{x}_{tl}, \hat{y}_{tl}, \hat{x}_{br}, \hat{y}_{br}]$ in Eq. (20)

4 EXPERIMENTS

4.1 Implementation Details

The network is trained on the COCO (Lin et al., 2014), ImageNet DET (Russakovsky et al., 2015), ImageNet VID (Russakovsky et al., 2015), LaSOT (Fan et al., 2019), and GOT-10k (Huang et al., 2019) training sets. The backbone parameters are initialized with ImageNet-pretrained ResNet-50. Our framework is trained for 50 epochs with 4000 iterations per epoch and 64 image pairs per batch on one Nvidia A100 GPU. The ADAM optimizer (Kingma Diederik and Adam, 2014) is employed with an initial learning rate of 0.001 and a decay factor of 0.5 for every eight epochs. Our method is implemented in Python using PyTorch.

4.2 Comparison with State-of-the-Art Trackers

We compare our proposed method with the recent state-of-the-art trackers published from 2019 to 2022 (SiamRPN++ (Li et al., 2019), DiMP-50 (Bhat et al., 2019), KYS (Bhat et al., 2020), SiamBAN (Chen et al., 2020), SiamAttn (Yu et al., 2020), SiamCAR (Guo et al., 2020), Ocean (Zhang et al., 2020), TrDiMP (Wang et al., 2021), SiamRN (Cheng et al., 2021), AR-DiMP50 (Yan et al., 2021), SiamGAT (Guo et al., 2021), AutoMatch (Zhang et al., 2021), TrTr (Zhao et al., 2021)), and MixFormer1K (Cui et al., 2022) on five tracking benchmarks, including VOT2018, UAV123, OTB100, and LaSOT.

In the VOT2018, the trackers are compared in terms of Accuracy (A), Robustness (R), and Expected

Average Overlap (EAO) metrics. A is the average overlap between the predicted and ground truth bounding boxes during successful tracking periods. R measures how many times the tracker loses the target (fails) during tracking. EAO is an estimator of the average overlap a tracker is expected to attain on a large collection of short-term sequences with the same visual properties as the given dataset. The one-pass evaluation criteria is used as defined in (Wu et al., 2015) to measure the tracking performance in terms of precision and success plots on OTB100, UAV123, and LaSOT datasets.

On VOT2018. The VOT2018 (Kristan et al., 2018) benchmark consists of sixty sequences with varying levels of difficulty, including many tiny, similar tracking objects. Detailed comparisons with the top-performing trackers are reported in Table 1. Our method achieves an EAO score of 0.492, significantly outperforming state-of-the-art methods on this metric. Compared with DiMP, our model achieves a performance gain of 5.3%. In comparison with the AR-DiMP50, we have a substantial improvement of 3.2% in EAO.

Table 1: Detail comparisons on VOT2018 with the state-of-the-art in terms of Accuracy (A), Robustness (R), Lost Number (LN), and Expected Average Overlap (EAO). Red and blue fonts indicate the top-2 trackers.

| Method | A(\uparrow) | R(\downarrow) | LN(\downarrow) | EAO(\uparrow) |
|--------------|-----------------|-------------------|--------------------|-------------------|
| Ours | 0.611 | 0.116 | 27.0 | 0.492 |
| SiamAttn | 0.630 | 0.159 | 34.0 | 0.470 |
| SiamRN | 0.595 | 0.131 | 28.0 | 0.466 |
| AR-DiMP50 | 0.642 | 0.159 | 34.0 | 0.460 |
| KYS | 0.603 | 0.143 | 30.5 | 0.458 |
| TrDiMP | 0.595 | 0.141 | 30.0 | 0.457 |
| SiamBAN | 0.590 | 0.178 | 38.0 | 0.447 |
| DiMP-50 | 0.597 | 0.152 | 32.5 | 0.439 |
| TrTr-Offline | 0.612 | 0.234 | - | 0.424 |
| SiamRPN++ | 0.600 | 0.234 | 50.0 | 0.415 |

In addition, we compare with state-of-the-art trackers in terms of EAO on several visual attributes, and the results are shown in Figure 3. Our approach scores top in motion change and illumination change and third in occlusion and camera motion. This demonstrates that our approach can overcome challenges.

On UAV123. UAV123 (Mueller et al., 2016) includes 123 low altitude aerial videos captured from a UAV. It features small objects, fast motions, occlusion, absent, and distractor objects. As demonstrated in Figure 4, the proposed method obtains 67.1% in terms of overall AUC score, which ranks two places, better than other trackers by a significant margin except

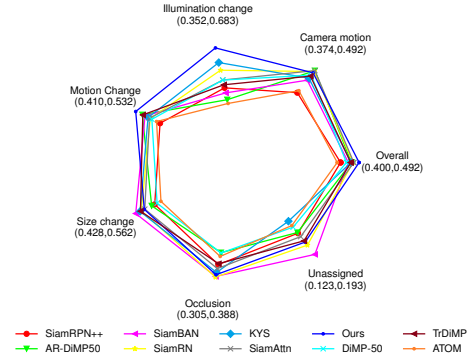


Figure 3: Comparison of EAO on VOT2018 for the following visual attributes: camera motion, illumination change, occlusion, size change, and motion change. Unassigned frames are those that do not relate to any of the five qualities. The parenthesis shows the EAO range of each tracker characteristic and overall.

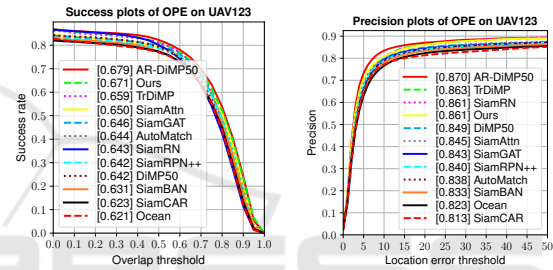


Figure 4: Comparisons on UAV123 in terms of success and precision plots of OPE. In the legend, the area-under-curve (AUC) and distance precision (DP) are reported in the left and right figures, respectively.

for AR-DiMP50 on success score, while AR-DiMP50 train with a mask option. The proposed methodology has a success score and a precision score higher than that of the DiMP50 model, which is 2.9% and 1.4%, respectively. Compared with the SiamAttn, a method developed from SiamRPN++, by adding the box refinement and mask branch, the TrackerLT model achieved a higher success score of 2.1% and a precision score greater than 1.6%.

On OTB100. OTB100 (Wu et al., 2015) contains 100 sequences in total and 11 challenge attributes, including illumination variation, out-of-plane rotation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-view, background clutter and low resolution. The proposed model has a success score of 0.701, higher than the remaining ten models, as shown in Figure 5. Compared with DiMP50, TrackerLT achieved higher success scores of 1.3%.

On LaSOT. LaSOT (Fan et al., 2019) is a recent large-scale dataset with high-quality annotations, which contains 280 for testing (2500 frames on average). We report the success and precision scores in the

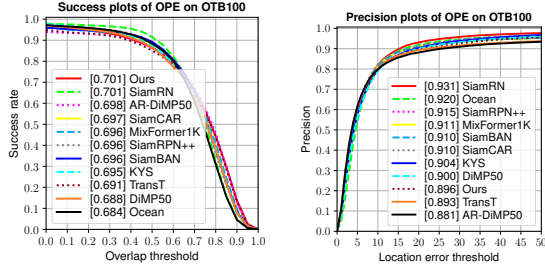


Figure 5: Comparison of success and precision plots on OTB100 with state-of-the-art methods. In the legend, the area-under-curve (AUC) and distance precision (DP) are reported in the left and right figures, respectively.

Table 2. This table shows that the proposed method obtains the best performance, better than other trackers by a significant margin except for AR-DiMP50, but AR-DiMP50 train with mask head. After applying TrackerLT to DiMP50 model, the success score improved by 3.5 percent compared to DiMP50. In addition, TrackerLT outperforms the two proposed models for 2021, AutoMatch and SiamGAT, by 2.2 percent and 6.5 percent for success scores and by 0.9 percent and 7.8 percent for precision ratings, respectively.

Table 2: A comparison of our method with other competitive approaches on the testing set of LaSOT in terms of success and precision metrics. The best two results are highlighted in red and blue, respectively.

| Tracker | Success(↑) | Precision(↑) |
|-----------|--------------|--------------|
| Ours | 0.604 | 0.608 |
| AR-DiMP50 | 0.602 | - |
| AutoMatch | 0.582 | 0.599 |
| DiMP50 | 0.569 | - |
| SiamGAT | 0.539 | 0.530 |
| Ocean | 0.516 | 0.526 |
| SiamRPN++ | 0.496 | 0.569 |

4.3 Ablation Study

4.3.1 Number of Transformer Layers

To compare the effect of numbers transformer layers with evaluation datasets. We tested with $L = 2$ and $L = 4$. As shown in Table 3, on VOT2018, when $L = 2$, the approach improved the EAO score by 2.6% compared to $L = 4$. Moreover, the network runs at 30 FPS, GPU Memory Usage is 3.238 GB and runs at 26 FPS, GPU Memory Usage is 3.31 GB when $L = 2$ and $L = 4$, respectively.

4.3.2 Type of Fusion

We have experimentally compared the results for 2 types of fusion, PW-Corr (Yan et al., 2021) and FFM.

Table 3: Quantitative comparison results of our method and its variants with different number of transformer layers on VOT2018. The best result is highlighted in red.

| Dataset | L | EAO(↑) | FPS(↑) | Memory |
|---------|-----|--------------|--------|----------|
| VOT2018 | 2 | 0.492 | 30 | 3.238 GB |
| | 4 | 0.466 | 26 | 3.310 GB |

As shown in Table 4, on VOT 2018, FFM with $L = 2$ has an EAO of 0.492, 0.6% higher than PW-Corr.

Table 4: Quantitative comparison results of our method and its variants with different type of fusion on VOT2018.

| Dataset | Fusion | EAO(↑) |
|---------|---------|--------------|
| VOT2018 | PW-Corr | 0.486 |
| | FFM | 0.492 |

4.4 Visualization

Figure 6 provides some representative visual results regarding the different methods. From top to bottom are videos from VOT2018, including nature, car1, and basketball. We can see that our TrackerLT module facilitates the tracker obtaining more precise bounding boxes than DiMP and AR-DiMP.



Figure 6: Visual comparison of TrackerLT and other methods. From left to right, we present the original prediction of the DiMP base tracker and refined results obtained by AR-DiMP, our TrackerLT. Color: Ground-Truth (GT), DiMP Base tracker (DiMP), AR-DiMP method (AR-DiMP) and our TrackerLT (TrackerLT).

5 CONCLUSIONS

In this research, we introduce a new neural network for visual tracking. We present a linear transformer module for enhancing features and an attention-based pixel-wise match module for combining features from two Siamese network branches. The new network

can significantly improve the DiMP tracker's robustness against illumination variation, background clutter, camera motion, and occlusion. Extensive testing on the VOT2018, UAV123, OTB100, and LaSOT benchmarks demonstrate that our technique provides state-of-the-art outcomes. In future work, we will extend our network by using the new fusions module and adding a mask branch prediction to boost the performance of trackers and address the challenges of fast motion, scale variation, and similar objects.

ACKNOWLEDGEMENTS

This research has been done under the research project TXTCN.22.02 of Vietnam National University, Hanoi.

REFERENCES

- Bhat, G., Danelljan, M., Gool, L. V., and Timofte, R. (2019). Learning discriminative model prediction for tracking. In *ICCV*.
- Bhat, G., Danelljan, M., Gool, L. V., and Timofte, R. (2020). Know your surroundings: Exploiting scene information for object tracking. In *ECCV*.
- Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., and Lu, H. (2021). Transformer tracking. In *CVPR*.
- Chen, Z., Zhong, B., Li, G., Zhang, S., and Ji, R. (2020). Siamese box adaptive network for visual tracking. In *CVPR*.
- Cheng, S., Zhong, B., Li, G., Liu, X., Tang, Z., Li, X., and Wang, J. (2021). Learning to filter: Siamese relation network for robust tracking. In *CVPR*.
- Cui, Y., Jiang, C., Wang, L., and Wu, G. (2022). Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*.
- 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 *CVPR*.
- Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., and Shen, C. (2021). Graph attention tracking. In *CVPR*.
- Guo, D., Wang, J., Cui, Y., Wang, Z., and Chen, S. (2020). Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.
- Huang, L., Zhao, X., and Huang, K. (2019). Got-10k: A large high-diversity benchmark for generic object tracking in the wild. In *TPAML*, volume 43. IEEE.
- Jiang, B., Luo, R., Mao, J., Xiao, T., and Jiang, Y. (2018). Acquisition of localization confidence for accurate object detection. In *ECCV*.
- Kingma Diederik, P. and Adam, J. B. (2014). A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Čehovin Zajc, L., Vojir, T., Bhat, G., Lukežić, A., Eldesokey, A., et al. (2018). The sixth visual object tracking vot2018 challenge results. In *ECCV Workshops*.
- 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 *CVPR*.
- 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*.
- Luvizon, D. C., Tabia, H., and Picard, D. (2019). Human pose regression by combining indirect part detection and contextual information. In *Computers & Graphics*, volume 85. Elsevier.
- Mueller, M., Smith, N., and Ghanem, B. (2016). A benchmark and simulator for uav tracking. In *ECCV*.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. In *IJCV*, volume 115. Springer.
- Schlag, I., Irie, K., and Schmidhuber, J. (2021). Linear transformers are secretly fast weight programmers. In *ICML*. PMLR.
- Tian, Z., Shen, C., Chen, H., and He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *ICCV*.
- 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*, volume 30.
- Wang, N., Zhou, W., Wang, J., and Li, H. (2021). Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*.
- Wu, Y., Lim, J., and Yang, M.-H. (2015). Object tracking benchmark. In *TPAMI*, volume 37.
- Yan, B., Zhang, X., Wang, D., Lu, H., and Yang, X. (2021). Alpha-refine: Boosting tracking performance by precise bounding box estimation. In *CVPR*.
- Yu, J., Jiang, Y., Wang, Z., Cao, Z., and Huang, T. (2016). Unitbox: An advanced object detection network. In *24th ACM international conference on Multimedia*.
- Yu, Y., Xiong, Y., Huang, W., and Scott, M. R. (2020). Deformable siamese attention networks for visual object tracking. In *CVPR*.
- Zhang, Z., Liu, Y., Wang, X., Li, B., and Hu, W. (2021). Learn to match: Automatic matching network design for visual tracking. In *ICCV*.
- Zhang, Z., Peng, H., Fu, J., Li, B., and Hu, W. (2020). Ocean: Object-aware anchor-free tracking. In *ECCV*.
- Zhao, M., Okada, K., and Inaba, M. (2021). Trtr: Visual tracking with transformer. *arXiv preprint arXiv:2105.03817*.