ENHANCED SEMANTIC ALIGNMENT IN TRANSFORMER TRACKING VIA POSITION LEARNING AND FORCE-DIRECTED ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

In the field of visual object tracking, one-stream pipelines have become the mainstream framework due to its efficient integration of feature extraction and relationship modeling. However, existing methods still face the issue of semantic misalignment: firstly, the interaction of position encoding between the two branches leads to a misalignment between feature semantics and position encoding; secondly, traditional attention mechanisms fail to distinguish between semantic attraction and repulsion among features, resulting in semantic misalignment when the model processes these features. To address these issues, we propose an Enhanced Semantic Alignment Transformer Tracker (ESAT) with position learning and force-directed attention. By leveraging self-supervised position loss (SPL), ESAT separately learns the absolute position encodings of the target and search branches, distinguishing the locations of various tokens and their positive or negative relationships, thereby enhancing the semantic consistency between position and features. Additionally, ESAT incorporates a repulsion-attraction mechanism applied to the self-attention module, named force-directed attention (FDA),simulating dynamic interactions between nodes to improve feature discrimination. Extensive experiments on multiple public tracking datasets show that our method outperforms many pipelines and achieves superior performance on five challenging benchmarks.

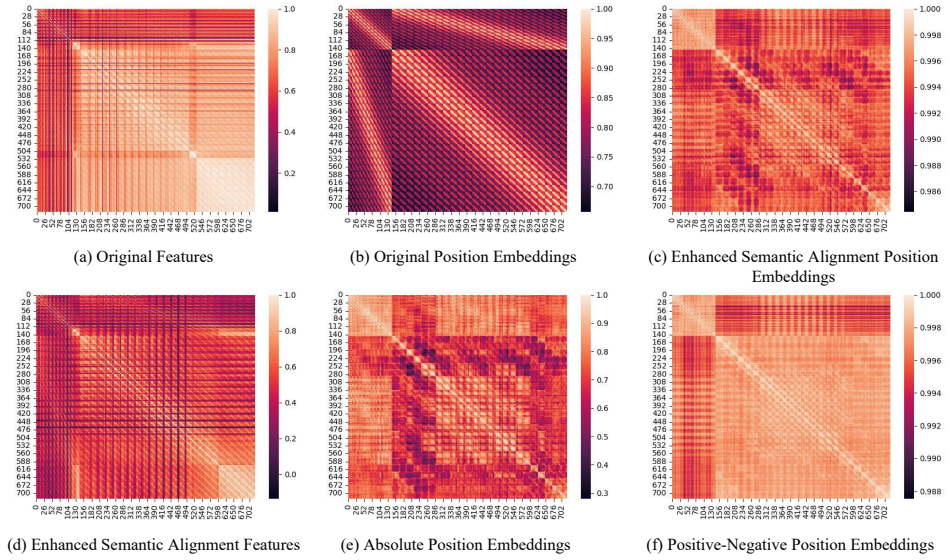
1 INTRODUCTION

Visual Object Tracking (VOT) has attracted increasing attention due to its broad applications in various fields, including traffic monitoring (Chandrakar et al., 2022), medical science (Bouget et al., 2017) and self-driving cars (Gao et al., 2019). Single Object Tracking (SOT) is a one of the popular category of VOT, which can be described as follows: Capture the target's appearance features in the first frame and maintain continuous tracking in subsequent frames (Soleimanitaleb & Keyvanrad, 2022). Notwithstanding many highly effective trackers have emerged (Bertinetto et al., 2016; Li et al., 2019), challenges such as occlusions and deformations continue to perplex numerous researchers, impinging on the accuracy and stability of tracking. In recent years, based on Transformer (Vaswani et al., 2017), Vision Transformer (ViT) (Dosovitskiy et al., 2020) with its superior attention mechanisms and capacity for global feature capture, have achieved remarkable success in VOT tasks (Chen et al., 2022; Lin et al., 2022; Hu et al., 2024).

Despite advancements in feature fusion, current prevailing trackers still have limitations in dealing with semantic alignment. **On the one hand**, self-attention cannot capture the ordering of input tokens, so incorporating positional encoding is particularly important. In VOT tasks, absolute positional encoding (Vaswani et al., 2017) is a widely adopted method. This method assigns distinct vectors to each position in the template and search token sequences, which may result in a semantic misalignment. Meanwhile, the template contains target feature information, while the search region includes potential target positions and background information. However, the differing semantic content and context of these areas may hinder the ability of model to effectively capture their relationships. **On the other hand**, traditional attention mechanisms in object tracking exhibit a significant limitation in feature weight allocation. Traditional methods indiscriminately sum the weights of various features, overlooking the complex interactions between them. This "one-size-fits-

all" approach is particularly inadequate given the semantic complexity, where feature relationships involving appearance, motion patterns, and background are interwoven and interdependent. For instance, in SOT tasks, certain features like the target's color may attract each other in specific contexts, while others, such as background interference, may repel due to semantic conflicts. Traditional attention mechanisms struggle to capture these subtle differences and dynamic relationships, leading to biases in the model's tracking accuracy and ultimately affecting its ability to discern the target's true state and motion trajectory.

To alleviate these problems, we propose a novel **Enhanced Semantic Alignment Transformer Tracker (ESAT)** with position learning and force-directed attention. Specifically, we introduce a novel self-supervised position loss function and a self attention mechanism incorporating repulsive and attractive forces. **(1)** Self-supervised position loss (SPL) allows each token to implicitly incorporate its positional information. We propose two types of positional labels: absolute positional labels and positive-negative positional labels. We input the features of the template and search regions outputted by the tracking model into two separate fully convolutional networks (FCN), each consisting of L stacked Conv-BN-ReLU layers. This module outputs the corresponding position information for each image patch. Based on this, we calculate the position losses using the two types of labels and the obtained positional information separately, distinguishing the positions of different tokens and their positive and negative values, to enhance the semantic consistency of positions and features. **(2)** Force-directed attention (FDA) define the repulsive and attractive forces, by calculating the difference between query (q) and key (k). Repulsive force reduces the attention weight between different features, while attractive force enhances the weight of similar features. By adjusting the attention matrix, the model considers the interaction between features in attention allocation, thereby reducing the problem of semantic misalignment.



(a) Original Features (b) Original Position Embeddings (c) Enhanced Semantic Alignment Position Embeddings

(d) Enhanced Semantic Alignment Features (e) Absolute Position Embeddings (f) Positive-Negative Position Embeddings

Figure 1: Visualization of the correlation between features of the template and search region, as well as their positional correlation. The lighter the color indicates the higher the correlation; conversely, the darker the color indicates the lower correlation. The visualization of correlations can be divided into four parts: template region self-correlation, search region self-correlation, and two sections representing the correlation between the template and the search image.

As depicted in Fig.1 (a), the original features have a smaller color distinction, while the features of the ESAT exhibit a greater color distinction, which indicates that the semantics of the model have been enhanced after applying ESAT shown in Fig.1 (d). Both template and search images start from 1, which can cause the most relevant areas to shift towards the beginning when the two regions interact, resulting in a misalignment of semantics, as illustrated in Fig.1 (b). Therefore, we apply self-supervised position learning to make the template most relevant to the target position in the search area. The learned absolute positional embeddings indicate that the most relevant regions for both are located in the middle, which corresponds to the areas where the template and the target po-

sition in the search region are most related. The positive-negative positional embeddings correspond to the absolute, highlighting the most relevant areas. By merging these two, we obtain the enhanced semantics alignment position embeddings as seen in Fig.1 (c).

To sum up, the main contributions of this work are threefold: **(1)** We propose a self-supervised position loss (SPL) and two positional labels: absolute and positive-negative positional labels, which mitigates the issue of semantic misalignment between template and search region. **(2)** We introduce force-directed attention (FDA) to adjust the attention weight matrix. FDA can enhance attention between similar features and weaken attention between different features to improve the semantics between features. **(3)** The above modules collaboratively form a comprehensive Enhanced Semantic Alignment Transformer Tracker (ESAT). Experiments on several widely used benchmarks demonstrate its outstanding performance and validate the efficiency and effectiveness of SPL and FDA.

2 Related Work

2.1 Transformer based Tracker

Transformer-based tracking models can be categorized into two distinct groups based on the framework and method of feature extraction they employ: CNN-Transformer based trackers and Fully-Transformer based trackers (Kugarajeevan et al., 2023).

CNN-Transformer based trackers (Chen et al., 2021b; Yan et al., 2021b) combine the Convolutional Neural Network (CNN) (Krizhevsky et al., 2012) and Transformer (Vaswani et al., 2017) to be a hybrid architecture. Following fully-convolutional siamese networks based trackers (Bertinetto et al., 2016; Li et al., 2019), they use two identical pipelines of CNNs to extracted the features of the target template and search region and then flat the features into vectors and pass this information to the Transformer which capture the similarity features of the target in the search region. Although CNN-Transformer based trackers utilize the attention mechanism, they still rely on CNN for feature extraction, which makes them to be difficult to capture global feature representations.

Fully-Transformer based trackers (Lin et al., 2022; Cui et al., 2022; Ye et al., 2022; Hu et al., 2024) are proposed to solve this problem, which are categorized into Two-stream Two-stage trackers and One-stream One-stage trackers. Two-stream Two-stage trackers use two identical and independent Transformer based tracking pipelines to extract features and then employ another Transformer network to find the relationships between these features. STMTrack (Fu et al., 2021) introduces a space-time memory network that integrates historical template and search features for better adapting to appearance variations during tracking. One-stream One-stage trackers like SimTrack (Chen et al., 2022) and OSTrack (Ye et al., 2022), using pretrained ViT as the backbone Transformer. In SimTrack, central concave window technology is developed to accurately capture target specific clues, which cropping the smaller areas of the template image, with the target in the middle, and then serialized into image markers. OSTrack points out that some search patches contain background information, which is not necessary in the tracking process, so an early candidate elimination strategy gradually discards the background regions in the search area during the inference process.

Although the dual stream method has achieved great success, the separation of feature extraction and relationship modeling has the following limitations. In this case, we use the One-stream One-stage method as as the baseline model.

2.2 ATTENTION MECHANISM IN TRANSFORMER TRACKING

The attention mechanism can establish rich global contextual dependencies, excelling at extracting edge features and distinguishing similar features, making it highly suitable for single object tracking tasks.

CNN-Transformer based trackers: STARK (Yan et al., 2021b) captures the feature dependency relationships between each element in the tracking sequence and enhances the original features using global contextual information, using multiple self attention layers to achieve information exchange and fusion between features. TransT (Chen et al., 2021b) replaces the traditional cross correlation with attention, using self attention and cross attention based feature enhancement module in the feature fusion stage.

Fully-Transformer based trackers: For Two-stream Two-stage methods, building on TransT (Chen et al., 2021b), SparseTT (Fu et al., 2022) proposes a sparse multi-head attention mechanism to improve the ability to differentiate between foreground and background. SwinTrack (Lin et al., 2022) explore to concatenate template and search features, and use multiple self-attention layers to facilitate feature interaction and fusion, reducing computational costs and enhancing model accuracy. CSWinTT (Song et al., 2022) proposed an attention module based on multi-scale cyclic moving windows, elevating pixel level attention to window level, which helps to maintain the integrity of the target and preserve more bit information. Force reduces the ambiguity of the target edge area. For One-stream One-stage methods, MixFormer (Cui et al., 2022) designs a set of Mixed Attention Modules based on the CVT (Wu et al., 2021a) structure to simultaneously extract and integrate features of the target and search region. Compared to CNN-Transformer based trackers, Fully-Transformer based method has a larger network capacity, stronger representation ability for its output features, and it exhibits greater robustness.

2.3 Positional Encoding

Transformer (Vaswani et al., 2017) relies on positional encoding (PE) to help it understand the spatial structure of images, as it does not utilize convolutional operations to extract local features. PE provides explicit spatial location information for each patch in the image. Each offering different advantages depending on the context of the task.

Absolute Position Encoding (APE) assigns a unique position vector to each element in the sequence or image, indicating its specific location. It provides a clear reference for the position of each element. There are several choices of APE. In Vision Transformer (ViT) (Dosovitskiy et al., 2020), the PE is generated by the fixed 1-D sinusoidal functions of different frequencies. At the same time, there are 2-D sinusoidal PE methods (Raisi et al., 2021; Wang & Liu, 2021) that incorporate information about the height and width of the image. Meanwhile, the APE can also be learnable, as it is randomly initialized and updated with the model's parameters during the training process.

Relative Position Encoding (RPE) encodes the relative distances or relationships between elements instead of focusing on the absolute position. It allows the model to understand how far apart elements are from each other. Based on the APE, RPE is added into the self-attention weight calculation. (Bello et al., 2019) firstly proposes a new relative positional encoding methods dedicated to 2D images, which is used in Swin-Transformer (Liu et al., 2021). (Wu et al., 2021b) further improves the 2-D RPE, called image RPE (iRPE), which integrates the modeling of directional relative distances as well as the interactions between queries and relative positional embeddings in self attention mechanism.

We build upon the previous method of explicitly combining positional information by using positional information as a supervised signal for the self-supervised training of the tracking model, allowing each encoded patch to implicitly contain its positional information.

3 Methodology

In this section, we introduce the framework of the proposed Enhanced Semantic Alignment Transformer Tracker (ESAT). Subsequently, we present self-supervised position loss and force-directed attention one by one.

3.1 Framework of ESAT

One-stream One-stage trackers integrate features more efficiently within a single Transformer, allowing simultaneous extraction and fusion. Therefore, we conduct research based on OSTrack (Ye et al., 2022). The pipeline of the tracker ESAT is shown in Fig.2. The whole framework can be divided into three parts:

Positional Label Generation. As shown in Fig.2 (a), we generate the corresponding absolute positional labels for each patch in the search image, while the absolute positional labels for the template image correspond to the central region of the search region. Next, we set the area corresponding to the ground truth box in the search image as the positional label of the template. For the positive-

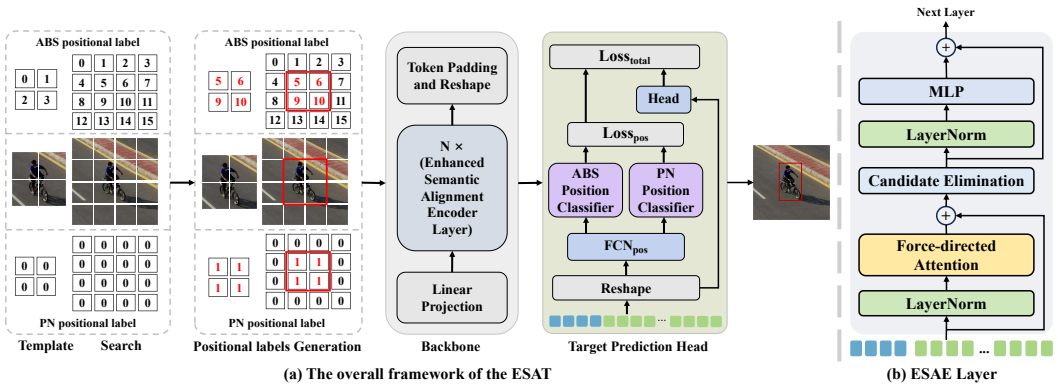


Figure 2: The pipeline of EAST.

negative positional labels, we assign a value of 1 to both the template region and the target area in the search region, while the background in the search region is set to 0.

Backbone. The input of the ESAT pipeline are template image z and search image x, and then split and flatten them into a series of patches $z_p \in \mathbb{R}^{N_z \times (3 \cdot P^2)}$ and $x_p \in \mathbb{R}^{N_x \times (3 \cdot P^2)}$, where N_z and N_x are respectively the number of template and search patches and $P \times P$ is the resolution of each image patch. Then, a Linear Projection Layer is used for generating the embeddings $E_z \in \mathbb{R}^{N_z \times D}$ of z_p and $E_x \in \mathbb{R}^{N_x \times D}$ of x_p, where D is the embedding dimension. Next, we input them into several subsequent Encoder Layers. Learnable 1D position embeddings P_z and P_x are added to the patch embeddings of the template and search region separately to produce the final template token embeddings $H_z \in \mathbb{R}^{N_z \times D}$ and search region token embeddings $H_x \in \mathbb{R}^{N_x \times D}$. After that, all these tokens are concatenated as a sequence with a length of $N_z + N_x$ and fed to an encoder. Each encoder layer updates the input tokens via a Force-directed Attention (FDA) block and a feed-forward network (FFN) as illustrated in Fig.2 (b).

Head and loss. The output template and search tokens from the last encoder layer are reshaped to the 2D feature maps according to their original spatial positions shown in Fig.2. We input the feature maps to fully convolutional networks (FCN) and concatenate the processed features, generating positional embeddings through linear layers. The search feature map is also taken as the input of to a convolutional head for target bounding box prediction. Finally, we calculate the loss and perform back-propagation to optimize the model parameters.

3.2 Self-Supervised Position Loss

Self-supervised position loss function (SPL) enables each token within a model to implicitly integrate its positional information, thereby enhancing the overall understanding of spatial relationships in the data. As shown in Fig.3, we introduce two distinct types of positional labels: absolute positional labels, which provide the exact location of each token within the input space, and positive-negative positional labels, which indicate whether a token is in a positive or negative position relative to its context.

To implement this, we take the features extracted from both the template and the search regions generated by the tracking model and feed them into two separate fully convolutional networks (FCNs). We denote the template and search feature maps as $E_T \in \mathbb{R}^{C \times H \times W}$ and $E_S \in \mathbb{R}^{C \times H \times W}$. Each FCN is designed with a structure comprising L stacked Conv-BN-ReLU layers, which allows the networks to effectively capture and process the spatial characteristics of the input features. After that, we concatenate the processed template and search features as the positional feature vector:

$$[\mathrm{p}_1; \mathrm{p}_2; \cdots; \mathrm{p}_{N_z+N_x}] = [\mathrm{FCN}(E_T); \mathrm{FCN}(E_S)], \tag{1}$$

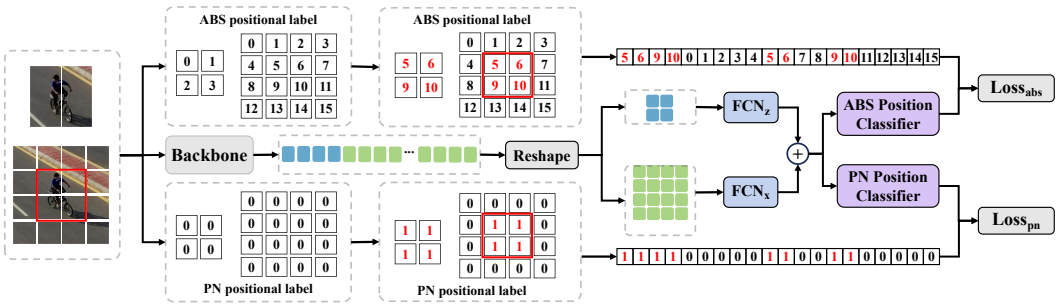


Figure 3: The process of self-supervised position loss.

Then, input the feature into absolute and positive-negative positional classifiers:

$$P_{\mathrm{p},y} = \frac{\exp\left(w^{\top}\mathrm{p}\right)}{\sum_{j=1}^{N}\exp\left(w_j^{\top}\mathrm{p}\right)}, \tag{2}$$

where p is positional feature; y is positional label; N is the length of the positional feature; $[\mathrm{w}_1;\mathrm{w}_2;\cdots;\mathrm{w}_N] \in \mathbb{R}^{D\times N}$ is the weights in the positional classifier, D denotes the dimension of the feature. These position embeddings not only contain information about the features themselves but also effectively incorporate positional information, enabling the model to better understand and utilize spatial relationships when performing tasks.

Once we have the positional information output from the FCNs, we proceed to compute the position losses using both types of labels. This computation is performed separately for the absolute positional labels and the positive-negative positional labels. By distinguishing the positions of different tokens and their associated positive and negative values, we aim to enhance the semantic consistency between the positions and the features represented by the tokens:

$$L_{abs} = -\frac{1}{n}\sum_{i=1}^{n}\log\left(P^{abs}_{\mathrm{p}_i,y_i^{abs}}\right), \tag{3}$$

$$L_{pn} = -\frac{1}{n}\sum_{i=1}^{n}\log\left(P^{pn}_{\mathrm{p}_i,y_i^{pn}}\right). \tag{4}$$

The integration of these position losses into the training process serves to reinforce the model's understanding of spatial relationships and improve its ability to accurately track objects across frames. By leveraging self-supervised learning mechanisms, our approach fosters a more robust and nuanced representation of positional information, ultimately leading to better performance in tasks that require precise localization and tracking.

3.3 Force-directed Attention

Force-Directed Attention (FDA) innovatively adjusts the distribution of attention among features by introducing the concept of force from physics, enabling the model to more effectively capture meaningful relationships when faced with complex features. FDA defines two interacting forces: repulsive force and attractive force by calculating the differences between queries and keys. Fig.4 illustrates an example of the FDA.

In FDA, the role of the repulsive force is to decrease the attention scores between different features. When the difference between the query and the key is large, the model applies repulsive force, causing these features to be suppressed in the attention distribution. This mechanism effectively reduces the model's focus on irrelevant features, thereby avoiding information interference and enhancing the model's sensitivity to important features. As Eq.5 shown, the repulsive force is calculated by determining the Euclidean distance between the query and the key, squaring it, and then scaling it with the hyperparameter δ to derive the repulsive force value $A_{repulsive}$. This value plays a suppressive

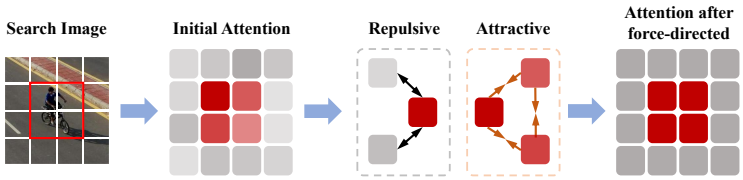


Figure 4: An illustration for Force-directed Attention Mechanism. The red blocks represent the target, while the gray blocks represent the background. Nodes with the same color indicate similar features, which attract each other, whereas nodes with different colors repel one another. Through force-directed modulation, the features of both colors are strengthened during the propagation process.

role in updating the attention scores, resulting in a decrease in the final attention scores A_{final} for features that are significantly different from the query.

$$A_{repulsive} = (\|q_1 - k_1\|_2^2 + \|q_2 - k_2\|_2^2) \cdot \delta, \tag{5}$$

where q_1 and q_2, as well as k_1 and k_2 represent different dimensions. We compute the differences across these dimensions separately to capture the feature relationships more comprehensively.

On the other hand, the attractive force enhances the attention scores between similar features. When the difference between the query and the key is small, the model applies attractive force, prompting an increase in the attention scores among similar features. The interaction between targets leads to a situation where the greater the repulsive force, the smaller the attractive force, and vice versa. This process applys an exponential function to the repulsive force, yielding an attractive force value $A_{attractive}$. The introduction of attractive force allows the model to better focus on semantically related features, thereby improving the interrelatedness of features and the efficiency of information transfer:

$$A_{attractive} = e^{-A_{repulsive}}. \tag{6}$$

Finally, by combining repulsive and attractive forces, the force-directed attention A_{final} is obtained:

$$A_{final} = A_{origin} + \alpha \cdot A_{attractive} - \beta \cdot A_{repulsive}, \tag{7}$$

where α and β are hyperparameter used to adjust attraction repulsion respectively.

Through the dynamic adjustment of repulsive and attractive forces, FDA can effectively reconstruct the attention matrix, enabling the model to better capture semantic information while considering the interactions between features. This mechanism reduces the problem of semantic misalignment among features, allowing FDA to help the model more accurately understand contextual relationships, thereby enhancing the effectiveness of task completion.

3.4 Target Prediction Head

During training, we use weighted focal loss (Law & Deng, 2018) to classify and perform bounding box regression by using L1 loss and generalized IoU loss (Rezatofighi et al., 2019). Combining the two position losses we designed, the overall loss function is:

$$L_{track} = L_{focal} + \lambda_{giou} L_{giou} + \lambda_{L_1} L_1 + \lambda_{abs} L_{abs} + \lambda_{pn} L_{pn}, \tag{8}$$

where $\lambda_{L_1} = 5$, $\lambda_{giou} = 2$, $\lambda_{abs} = 0.01$ and $\lambda_{pn} = 0.1$ are the regularization parameters.

4 Experiments

4.1 Implementation

The experiment is conducted on a server with 8 NVIDIA A100 GPUs. We implement our Enhanced Semantic Alignment Transformer Tracker using Python 3.8 and PyTorch 1.9.

Table 1: Comparison with state-of-the-art trackers on five popular benchmarks. The best two results are shown in **bold font**.

Method	Year	LaSOT			GOT-10K			TNL2K		TrackingNet			$\text{LaSOT}_{\text{ext}}$		
		AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$	AUC	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P
SiamPRN++ (Li et al., 2018)	CVPR19	49.6	56.9	49.1	-	-	-	41.3	41.2	73.3	80.0	69.4	34.0	41.6	39.6
DiMP (Bhat et al., 2019)	ICCV19	56.9	65.0	56.7	61.1	71.7	49.2	44.7	43.4	74.0	80.1	68.7	39.2	47.6	45.1
ATOM (Danelljan et al., 2019)	CVPR19	51.5	57.6	-	-	-	-	40.1	39.2	-	-	-	-	-	-
SiamR-CNN (Voigtlaender et al., 2019)	CVPR20	64.8	72.2	-	64.9	72.8	59.7	-	-	81.2	85.4	80.0	-	-	-
Ocean (Zhang & Peng, 2020)	ECCV20	56.0	65.1	56.6	61.1	72.1	47.3	38.4	37.7	-	-	-	-	-	-
TrDiMP (Wang et al., 2021a)	CVPR21	63.9	-	61.4	67.1	77.7	58.3	-	-	78.4	83.3	73.1	-	-	-
TransT (Chen et al., 2021a)	CVPR21	64.9	73.8	69.0	67.1	76.8	60.9	-	-	81.4	86.7	80.3	-	-	-
AutoMatch (Zhang et al., 2021)	ICCV21	58.3	-	59.9	65.2	76.6	54.3	47.2	43.5	76.0	-	72.6	-	-	-
STARK (Yan et al., 2021a)	ICCV21	67.1	77.0	-	68.8	78.1	64.1	-	-	82.0	86.9	-	-	-	-
TransInMo (Guo et al., 2022)	CVPR22	65.7	76.0	70.7	-	-	-	52.0	52.7	-	-	-	-	-	-
AiATrack (Gao et al., 2022)	ECCV22	69.0	79.4	73.8	69.6	80.0	63.2	-	-	82.7	87.8	80.4	-	-	-
SwinTrack-B (Lin et al., 2022)	NeurIPS22	71.3	-	76.5	72.4	80.5	67.8	55.9	57.1	82.5	87.0	80.4	49.1	-	55.6
CiteTracker (Li et al., 2023)	ICCV23	69.7	78.6	75.7	**74.7**	**84.3**	**73.0**	57.7	**59.6**	**84.5**	**89.0**	**84.2**	-	-	-
MixFormerV2 (Cui et al., 2024)	NeurIPS23	70.6	80.8	76.2	-	-	-	57.4	58.4	83.4	88.1	81.6	50.6	-	56.9
MATTrack (Zhao et al., 2023)	CVPR23	67.8	77.3	-	67.7	78.4	-	51.3	-	81.9	86.8	-	-	-	-
OmniTracker (Wang et al., 2023)	CVPR23	69.1	77.3	75.4	-	-	-	51.3	-	83.4	86.7	82.3			
GRM (Gao et al., 2023)	CVPR23	69.9	79.3	75.8	73.4	82.9	70.4	-	-	84.0	88.7	83.3			
DATr (Zhao et al., 2024)	WACV24	71.0	80.7	77.5	74.2	84.1	71.1	-	-	-	-	-	**51.8**	**62.7**	**59.0**
STCFormer (Hu et al., 2024)	AAAI24	**71.5**	**81.5**	**78.0**	74.3	84.2	**72.6**	57.7	59.0	-	-	-	**52.0**	**63.0**	**59.6**
AQAT (Xie et al., 2024)	CVPR24	71.4	**81.9**	**78.6**	73.8	83.2	72.1	**57.8**	59.4	-	-	-	51.2	62.2	58.9
OSTrack (Ye et al., 2022)	ECCV22	71.1	81.1	77.6	73.7	83.2	70.8	55.9	-	83.9	88.5	82.0	50.5	61.3	57.6
ESAT	Ours	**71.6**	81.4	**78.0**	**74.7**	**84.8**	71.5	**58.2**	**59.9**	**84.2**	**88.7**	**83.4**	51.2	62.2	58.4

Model. Similar to OSTrack, our model consists of 12 sequential encoder layers, initialized with a MAE pre-trained model for the backbone network, where the search area pixels are 384×384 and the template pixels are 192×192. .

Training. We train our model with following datasets: COCO (Lin et al., 2014), LaSOT (Fan et al., 2018), GOT-10k (Huang et al., 2018) and TrackingNet (Müller et al., 2018). During the training process, we set the batch size to 16, the initial learning rate of the backbone to 4×10^{-5}, the learning rate of other parameters to 4×10^{-4}, and weight decay to 10^{-4}. The total number of training epochs is 300, and after 240 epochs, we decrease the learning rate by a factor of 10. The model is trained with AdamW optimizer.

Inference. During inference, we update the parameters normalization according to loss. The benchmarks we set for test are LaSOT, GOT-10k, TNL2K, TrackingNet and $\text{LaSOT}_{\text{ext}}$.

4.2 Results and Comparisons

To verify the efficacy of the proposed model, we compare them with 20 state-of-the-art approaches on five different benchmarks. Results are shown in Tab. 1.

LaSOT (Fan et al., 2018). LaSOT focuses on long-term tracking scenarios, providing a large-scale dataset that includes over 1400 video clips and more than 3.5 million frames, covering 70 categories. Our ESAT performs 0.5% higher than baseline OSTrack-384 in AUC as well as 0.3% higher in P_{Norm} and 0.4% higher in P, which means our method is suitable in long-term video sequences.

GOT-10k (Huang et al., 2018). GOT-10k is a dataset for general object tracking, including over 10000 video clips. It takes the average overlap (AO) and the success rate (SR) at overlap thresholds 0.5 and 0.75 as the evaluation metrics. We obtain improvements of the AO, $\text{SR}_{0.5}$, $\text{SR}_{0.75}$ of 1.0%, 1.6%, 0.7%, respectively. And we have a significant advantage over other methods.

TNL2K (Wang et al., 2021b). TNL2K is a new dataset for natural language guided tracking, which contains 700 high diversity sequences and introduces several adversarial samples and thermal images to improve the generality of tracking evaluation. Therefore, TNL2K is a challenging benchmark currently. We created a new state-of-art on it with an AUC of 58.2%, which is 2.3% higher than our baseline OSTrack-384. And our scores in P also exceed other state-of-the-art tracking algorithms.

TrackingNet (Muller et al., 2018). TrackingNet is a large-scale dataset aimed at evaluating the performance of target tracking algorithms in real-world scenarios. It contains over 30000 video clips, covering a variety of different scenes and targets. On TrackingNet, our method outperforms

Table 2: Quantitative comparison results of tracker with different components. The best results are shown in **bold font**.

OSTrack-384	ABS Position Loss	PN Position Loss	FDA	GOT-10K	TNL2K	LaSOT	Speed(fps)	MACs(G)	Params (M)
✓				73.7	55.9	71.1	114.4	48.4	92.1
✓	✓			74.0	58.0	71.3	106.6	50.9	99.0
✓		✓		74.3	57.9	71.3	106.8	50.9	99.0
✓			✓	73.6	57.6	71.3	96.2	48.4	92.1
✓	✓	✓		**74.7**	58.0	71.5	107.5	50.9	99.0
✓	✓		✓	73.9	57.9	71.4	82.3	50.3	94.8
✓		✓	✓	74.2	57.7	71.3	81.7	50.3	94.8
✓	✓	✓	✓	**74.7**	**58.2**	**71.6**	83.2	50.3	94.8

Table 3: Effect of different weights for ABS position loss and PN position loss. The best results are shown in **bold font**.

ABS Position Loss	PN Position Loss	GOT-10K	TNL2K	LaSOT
0.001	-	73.9	57.6	71.1
0.01	-	**74.0**	58.0	**71.3**
0.1	-	73.5	**58.1**	70.9
-	0.01	73.8	**57.9**	71.2
-	0.1	**74.3**	**57.9**	**71.3**
-	1	73.6	57.6	71.1
0.01	0.01	73.5	57.8	71.1
0.01	0.1	**74.7**	**58.0**	**71.5**
0.1	0.01	73.6	57.8	70.8
0.1	0.1	74.1	58.0	71.3

favorably against most other trackers, achieves 84.2% in success score, 88.7% in P_{Norm} score and 83.4% in P score, overtaking most previously published trackers.

$LaSOT_{ext}$ (Fan et al., 2020). $LaSOT_{ext}$ is an extended target tracking dataset based on the LaSOT dataset, which includes 150 additional sequences of 15 object classes. Due to its late release, there were relatively few results, but we still have a significant improvement compared to the baseline. Our AUC, P and P_{Norm} have increased by 0.7%, 0.8% and 0.9%, respectively.

4.3 Ablation Study

The Gains of Each Module. We explored the situations when three modules are used separately or in combination on three benchmarks to judge the exact impact of each part. More specifically, all three parts have a significant improvement on TNL2K and LaSOT whether used alone or in combination. The improvement on GOT-10K is evident when using both types of position loss functions. Tab. 2 shows the results, which prove that each module has made a contribution to the improvement of performance.

Speed and Size. Our ESAT runs at a reduced speed compared to the baseline. As shown in the Tab. 2, the parameters of our ESAT are 94.8M, which are 2.7M larger than the original OSTrack-384. As for multiply-accumulate computations (MACs), ours are 50.3G, about 1.9G larger than the baseline. Experiments show that there is not too much computational burden of our method. The addition of FDA will lead to a decrease in running speed and a little increase in the computational burden of parameters and MACs, while the addition for position learning information almost cause no extra burden in speed in comparison to the baseline OSTrack-384.

Effect of weights on each position loss. We try different weights for λ for each loss we design. The range of testing weights of each loss is approximately determined by the scale of their values. According to Tab. 3, results show that 0.01 and 0.1 are best for absolute position loss and positive-negative position loss, whether used separately or together.

Table 4: Effect of values on hyperparameter α and β. The best results are shown in **bold font**.

α&β	1	0.1	0.01
LaSOT	71.1	**71.6**	71.4
TNL2K	57.8	**58.2**	58.0

Effect of values on hyperparameter α and β. We also try different values on hyperparameter α and β for $A_{attractive}$ and $A_{repulsive}$. Tab. 4 shows that the best result is 0.01.

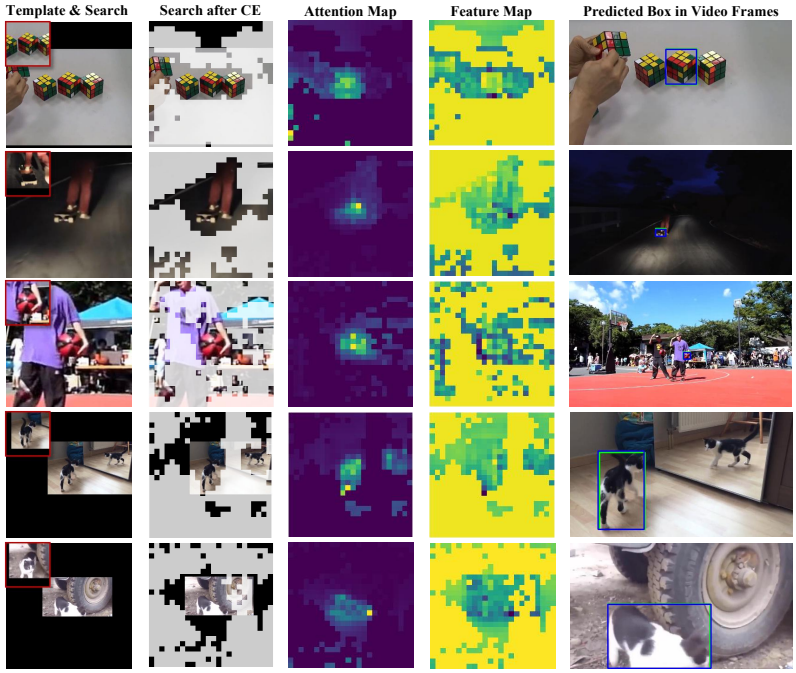


Figure 5: Visualization of the tracking process. The first column is the search images (the big ones) and their template images (small ones on upper left corner). The second column shows the search images after the early candidate elimination process of OSTrack. The third column shows attention map for corresponding search image. The fourth column shows our feature map for this search. The fifth column is the tracking results on corresponding frames. The green rectangles are groundtruth and the blue rectangles are our predicted box.

Visualization. Fig.5 exhibits some examples of our realtime tracking, and we can see that ESAT is robust in most cases and is able to deal with many challenges. It is able to catch small target (row 3) and distinguish the target from objects that look similar like it (row 1 and row 4). It can also track accurately when there are changes in lighting conditions (row 2) and deal with deformation of the target (row 5).

5 Conclusion

In this paper, we propose a novel Enhanced Semantic Alignment Transformer Tracker (ESAT) to alleviate the problem of semantic misalignment. Specifically, we design the self-supervised position loss (SPL) and force-directed attention (FDA). SPL learns absolute positional encoding of both target and search regions, and distinguish the target and background through positive-negative labels. FDA uses repulsive and attractive forces to adjust attention and enhance semantic relationships among features. The two modules cooperate to improve semantic alignment. By embedding these modules into ESAT, extensive and quantitative experiments demonstrate the effectiveness of our approach. We hope that our work can provide new insights into addressing challenges related to semantic misalignment in Transformer-based trackers.

REFERENCES

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3286–3295, 2019.

Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pp. 850–865. Springer, 2016.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6181–6190, 2019.

David Bouget, Max Allan, Danail Stoyanov, and Pierre Jannin. Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *Medical image analysis*, 35:633–654, 2017.

Ramakant Chandrakar, Rohit Raja, Rohit Miri, Upasana Sinha, Alok Kumar Singh Kushwaha, and Hiral Raja. Enhanced the moving object detection and object tracking for traffic surveillance using rbf-fdlnn and cbf algorithm. *Expert Systems with Applications*, 191:116306, 2022.

Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In *European Conference on Computer Vision*, pp. 375–392. Springer, 2022.

Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8122–8131, 2021a.

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*, pp. 8126–8135, 2021b.

Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13608–13618, 2022.

Yutao Cui, Tianhui Song, Gangshan Wu, and Limin Wang. Mixformerv2: Efficient fully transformer tracking. *Advances in Neural Information Processing Systems*, 36, 2024.

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*, pp. 4660–4669, 2019.

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.

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. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5369–5378, 2018.

Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129:439–461, 2020.

Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13774–13783, 2021.

Zhihong Fu, Zehua Fu, Qingjie Liu, Wenrui Cai, and Yunhong Wang. Sparsett: Visual tracking with sparse transformers. *arXiv preprint arXiv:2205.03776*, 2022.

Ming Gao, Lisheng Jin, Yuying Jiang, and Baicang Guo. Manifold siamese network: A novel visual tracking convnet for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1612–1623, 2019.

Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *European Conference on Computer Vision*, pp. 146–164. Springer, 2022.

Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18686–18695, 2023.

Mingzhe Guo, Zhipeng Zhang, Heng Fan, Liping Jing, Yilin Lyu, Bing Li, and Weiming Hu. Learning target-aware representation for visual tracking via informative interactions. *arXiv preprint arXiv:2201.02526*, 2022.

Kun Hu, Wenjing Yang, Wanrong Huang, Xianchen Zhou, Mingyu Cao, Jing Ren, and Huibin Tan. Sequential fusion based multi-granularity consistency for space-time transformer tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12519–12527, 2024.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:1562–1577, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Janani Kugarajeevan, Thanikasalam Kokul, Amirthalingam Ramanan, and Subha Fernando. Transformers in single object tracking: An experimental survey. *IEEE Access*, 2023.

Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 734–750, 2018.

Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4277–4286, 2018.

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*, pp. 4282–4291, 2019.

Xin Li, Yuqing Huang, Zhenyu He, Yaowei Wang, Huchuan Lu, and Ming-Hsuan Yang. Citetracker: Correlating image and text for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9974–9983, 2023.

Liting Lin, Heng Fan, Zhipeng Zhang, Yong Xu, and Haibin Ling. Swintrack: A simple and strong baseline for transformer tracking. *Advances in Neural Information Processing Systems*, 35:16743–16754, 2022.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

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*, pp. 10012–10022, 2021.

Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. *ArXiv*, abs/1803.10794, 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 *Proceedings of the European conference on computer vision (ECCV)*, pp. 300–317, 2018.

Zobeir Raisi, Mohamed A Naiel, Georges Younes, Steven Wardell, and John Zelek. 2lspe: 2d learnable sinusoidal positional encoding using transformer for scene text recognition. In *2021 18th Conference on Robots and Vision (CRV)*, pp. 119–126. IEEE, 2021.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666, 2019.

Zahra Soleimanitaleb and Mohammad Ali Keyvanrad. Single object tracking: A survey of methods, datasets, and evaluation metrics. *arXiv preprint arXiv:2201.13066*, 2022.

Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8791–8800, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and B. Leibe. Siam r-cnn: Visual tracking by re-detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6577–6587, 2019.

Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Xiyang Dai, Lu Yuan, and Yu-Gang Jiang. Omnitracker: Unifying object tracking by tracking-with-detection. *arXiv preprint arXiv:2303.12079*, 2023.

Ning Wang, Wen gang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1571–1580, 2021a.

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*, pp. 13763–13773, 2021b.

Zelun Wang and Jyh-Charn Liu. Translating math formula images to latex sequences using deep neural networks with sequence-level training. *International Journal on Document Analysis and Recognition (IJDAR)*, 24(1):63–75, 2021.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22–31, 2021a.

Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10033–10041, 2021b.

Jinxia Xie, Bineng Zhong, Zhiyi Mo, Shengping Zhang, Liangtao Shi, Shuxiang Song, and Rongrong Ji. Autoregressive queries for adaptive tracking with spatio-temporaltransformers. *arXiv preprint arXiv:2403.10574*, 2024.

Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10428–10437, 2021a.

Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10448–10457, 2021b.

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European conference on computer vision*, pp. 341–357. Springer, 2022.

Zhipeng Zhang and Houwen Peng. Ocean: Object-aware anchor-free tracking. *ArXiv*, abs/2006.10721, 2020.

Zhipeng Zhang, Yihao Liu, Xiao Wang, Bing Li, and Weiming Hu. Learn to match: Automatic matching network design for visual tracking. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13319–13328, 2021.

Haojie Zhao, Dong Wang, and Huchuan Lu. Representation learning for visual object tracking by masked appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18696–18705, 2023.

Jie Zhao, Johan Edstedt, Michael Felsberg, Dong Wang, and Huchuan Lu. Leveraging the power of data augmentation for transformer-based tracking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6469–6478, 2024.