2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# SDTrack:Spatially decoupled tracker for visual tracking

Zihao Xia[1], Xin Bi[1], Baojie Fan*, and Zhiquan Wang

***Abstract*— Recent models based on encoder-decoder architecture have shown excellent performance in visual object tracking. The encoder models the global spatiotemporal feature correlation between the template and the search regions, while the decoder learns query embeddings to predict the spatial location of the target. However, in previous methods, decoders are query-shared, which may lead to suboptimal results. We observe that different regions in the visual feature map are suitable for performing different tasks. Salient regions in object provide important information for classification task, while the boundaries around it are more beneficial for box localization task. We therefore propose a spatially decoupled tracker called SDTrack. The tracker contains a query selection module that we carefully design to select appropriate queries for both classification and regression tasks. We divide the cross-attention module in the decoder and add the box-to-pixel relative position offset (BoxRPB) term to the cross-attention, so that the attention is more focused on the respective areas of interest while introducing smaller overhead. Finally, we propose an alignment loss to solve the misalignment problem between accurate classification and precise localization, further improving tracking performance. Through extensive experiments, we demonstrate that SDTrack achieves new SOTA performance on multiple benchmarks compared to previous work, while running at real-time speeds.**

## I. INTRODUCTION

Visual object tracking is one of the core areas of computer vision research. Its main task is to accurately lock and track specific target in real time through in-depth analysis of continuous image sequences. In recent years, thanks to the rise of Transformer [1] technology, visual object tracking technology has made major breakthroughs. Transformer's unique attention mechanism enables it to capture the association between any two points in the input sequence, thereby handling long-distance dependencies more effectively and utilizing global context information. This feature enables Transformer to exhibit better positioning and tracking capabilities when tracking targets that undergo large-scale changes or frequently enter and exit the field of view.

*This work is supported by the National Natural Science Foundation of China ( No. 62473205, U2013210, 62103388), and the young and middle-aged leading scholar in Qinglan Project by Jiangsu Province

[1]These authors contributed to the work eqully.

Zihao Xia is with College of Automation and College of ArtificialIntelligence, Nanjing University of Posts and Telecommunications xzhfirst0430@163.com

Xin Bi is with College of Automotive Studies, Tongji University bixin@tongji.edu.cn

*corresponding authors. Baojie Fan is with the College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications jobfbj@gmail.com

Zhiquan Wang is with College of Automation and College of ArtificialIntelligence, Nanjing University of Posts and Telecommunications czember@163.com

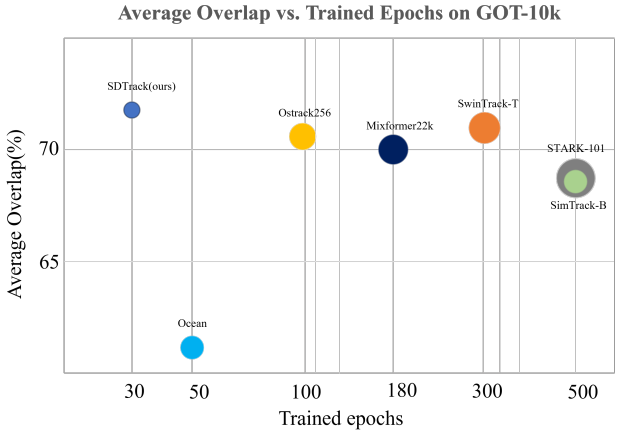


Fig. 1. Comparison of our SDTRack with other trackers on GOT-10k benchmark in terms of trained epochs following the official one-shot protocol. The bubble size represents the relative GFLOPs. Our SDTRack sets up a best trade off among accuracy, trained epochs and running GFLOPs.

The core process of the Transformer-based tracker mainly covers three links: (i) a neural network to extract features of templates and search images; (ii) an attention mechanism to optimize feature matching; and (iii) accurately locate the target through a carefully designed head. Recently, as the flexibility of the attention mechanism has been fully utilized, the first two links have been integrated through a unified architecture [2]–[4], effectively improving performance and efficiency. However, despite these trackers achieving significant improvements in performance, the demands on time and computing resources are still high. Inspired by [5], we decide to adopt the encoder-decoder architecture as shown in the figure 2(a). The decoder is able to process feature sequences, and it focuses on the preserved tokens processed by the encoder, thus avoiding redundant computations. However, [5] simply selects as input the top k highest-scoring tokens from the tokens processed by the encoder through the linear layer to the decoder. This approach ignores differences in token spatial locations, and using these markers together for localization and classification may lead to semantic mismatches that affect tracking accuracy. We found that features at different locations within an object contribute differently to classification and localization. For example, the salient area of an object provides key information for classification, while the boundary area of an object is more conducive to bounding box localization.

Therefore, we propose a decoupled design scheme for the decoder as shown in Figure 2(b). Unlike a completely independent two-branch design, we split the cross-attention block

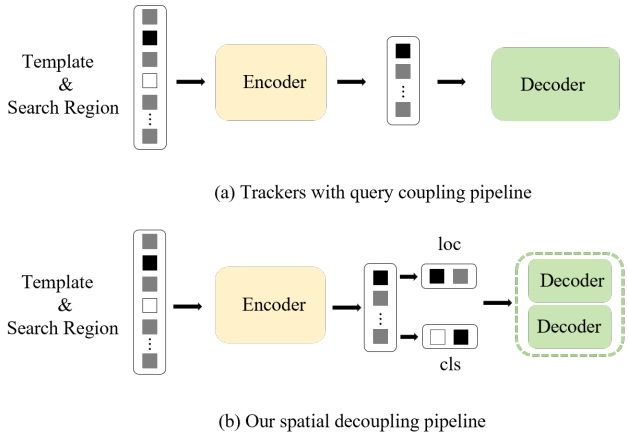


Fig. 2. (a)Trackers directly uses the search area token output by the encoder to the decoder; (b)Our SDTracker decouples the search token and decoder to achieve more precise positioning.

in the decoder into two branches, enabling the classification and localization tasks to target different regions of the visual feature map for query matching. In each cross-attention block, we include BoxRPB terms to ensure that attention is focused on the target and bounding boxes. Both branches share the self-attention layer to facilitate collaboration between them. With this decoupling approach, our approach achieves higher tracking performance. After separating the classification and positioning branches, how to initialize content and location embeddings becomes a key issue. To solve this problem, we design a query selection module that is able to learn task-specific queries based on anchor boxes. In addition, we also noticed the misalignment problem between classification and localization, that is, the confidence of classification is high but the localization is not accurate enough. To solve this problem, we further propose an alignment loss to learn the consistency between the two tasks based on [6].

Our extensive experiments validate the effectiveness and efficiency of our SDTRack, achieving higher performance with lower GFLOPs as shown in Figure 1. Our main contributions are summarized below:

- We design the query selection module to initialize queries for classification and localization branches so that they can better match different regions of objects.
- We design a feature decoupling method that divides cross-attention blocks in the decoder while adding BoxRPB term, which significantly improves tracking accuracy and introduces less overhead.
- We propose alignment loss to guide consistency between high classification confidence and precise localization, and achieved state-of-the-art performance on multiple benchmark datasets.

## II. RELATED WORKS

### A. Visual Tracking Paradigms

In recent years, Siamese trackers [7] have gradually received widespread attention. These trackers usually adopt a dual-stream architecture to handle feature extraction of templates and search regions separately. In order to characterize the interaction between the two streams, they also introduce specific related modules. Recently, with the introduction of Transformer, trackers have made significant progress in the exploration of feature interactions, providing strong support for the development of more advanced tracking algorithms, and gradually becoming the first choice for many high-performance trackers [8]–[11]. In order to further enhance the interaction between features, some research has begun to focus on modeling cross-relationships within the backbone network. Inspired by these explorations, some researchers began to design one-stream tracking frameworks to jointly encode templates and search regions. However, similar to previous two-stream methods, these single-stream methods usually treat the search region as a whole, resulting in the template interacting with all parts within the search region. When feature representation is not perfect in modeling cross-relationships, it may cause confusion between target and background.

### B. Tracking Paradigm for encoder-decoder Architecture

[12] proposed an end-to-end object detector called DETR, which is based on an encoder-decoder architecture and abandons traditional anchor design and manual components such as NMS. Inspired by DETR, STARK [11] transforms object tracking into a bounding box prediction problem and adopts an encoder-decoder converter to deal with it. In STARK, the encoder is responsible for capturing the global spatiotemporal feature dependence between the target and the search region. On the other hand, Detrack [5] also adopts an encoder-decoder architecture similar to DETR and is applied to visual object tracking without convolutional heads, thereby maintaining efficient computation of the sparse backbone. However, Detrack does not take into account the problem of sharing queries in classification and localization tasks, and sharing cross-attention between different queries that may lead to performance degradation. To overcome these limitations, we choose to separate the query before the decoder and perform feature solution learning in the decoder. In this way, we can select the most appropriate query for the classification and localization branches, ensure that the cross-attention of each branch is focused on its region of interest, and reduce feature conflicts. This method significantly improves tracking accuracy.

### C. Feature Decoupling Method

In the era of convolutional neural networks, the misalignment between classification and localization branches has been extensively studied in the field of object detection. For example, IoUNet found that features that generate high classification scores often predict only coarse bounding boxes. To solve this problem, IoUNet introduces an additional prediction head that estimates the intersection of unions (IoU) as the confidence of the localization, which is subsequently combined with the classification confidence to derive the final classification score. Another technology, two-head

R-CNN, decomposes the originally interrelated classification and localization tasks into two independent branches. There is also a method called TSD, which decouples classification and localization by spatially separating gradient flows. In addition, some innovative label assignment methods, such as TOOD and MuSu, propose the concept of anchor alignment metric, which can be integrated into the sample assignment and loss function to dynamically promote the consistency between high classification confidence and precise localization. . Although these methods have achieved remarkable results in convolutional neural networks, their application effects in encoder-decoder architectures similar to DETR still require further research and verification.

## III. METHOD

This section details our SDTrack approach. First, we describe the proposed model architecture, then, introduce how to spatially decouple and disentangle feature learning for our tracker, and finally elaborate on our introduced BoxRPB term and the proposed alignment loss.

### A. Overview

The overall architecture of SDTrack is shown in Figure 3. The tracker receives as input a pair of images, namely a template image $t \in R^{H_t \times W_t \times 3}$ and a search region image $s \in R^{H_s \times W_s \times 3}$. First, these images are divided into $N_t$ and $N_s$ non-overlapping image patches, respectively, and each image patch has a resolution of $P \times P$. Among them $N_t = H_t W_t / P^2$ and $N_s = H_s W_s / P^2$. Subsequently, these image patches are converted into template patch embedding $\mathbf{E}_t \in \mathbb{R}^{N_t \times C}$ and search patch embedding $\mathbf{E}_s \in \mathbb{R}^{N_s \times C}$ through linear projection operations, where $C$ is the dimensional size of the embedding. To incorporate spatial information, two learnable position embeddings $E_t$ and $E_s$ are added to the template block embedding and search block embedding respectively. Next, all these tokens are concatenated into a sequence of length $N_t + N_s$ and fed to the encoder for processing. In the encoder, each encoder layer contains a multi-head attention (MHA) block and a feed-forward network (FFN) for updating input tokens. Mathematically, the operation of the $l$-th encoder layer can be expressed as:

$$\begin{aligned} \boldsymbol{q} = \boldsymbol{k} = \boldsymbol{v} &= [\mathbf{E}_t^l; \mathbf{E}_s^l], \\ [\mathbf{E'}_t^l; \mathbf{E'}_s^l] &= [\mathbf{E}_t^l; \mathbf{E}_s^l] + \mathrm{MHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}), \end{aligned} \tag{1}$$

and the final output can be described as:

$$[\mathbf{E}_z^{l+1}; \mathbf{E}_x^{l+1}] = [\mathbf{E'}_z^l; \mathbf{E'}_x^l] + \mathrm{FFN}([\mathbf{E'}_z^l; \mathbf{E'}_x^l]), \tag{2}$$

where $\mathbf{E}_t^l$ and $\mathbf{E}_s^l$ are the input tokens of the $l$-th encoder layer, and $[;]$ represents the cascade operation. We use $\boldsymbol{q}$, $\boldsymbol{k}$ and $\boldsymbol{v}$ to represent the queries, keys and values provided to the multi-head attention block. Then, the output search tokens from the last encoder layer are decoupled from the template tokens, followed by further decoupling through our query selection module (see Section III-C for details), and finally the decoupled classification query and positioning query is fed into our spatially decoupled decoder for subsequent target localization.

### B. Disentangled Feature Learning

As analyzed before, in trackers based on the encoder-decoder architecture, the inherent conflicts caused by sharing queries between different tasks and sharing cross-attention operations between different queries significantly restrict its performance. In order to alleviate this problem, we adopted the method of task disentanglement, specifically focusing on two aspects: one is disentanglement feature learning, and the other is the disentanglement query generation process. Among them, the operation of disentangled feature learning can be described as:

$$\begin{aligned} \hat{Q} = \{&\text{self-att}(\text{cat}(\mathrm{Q_c}, \mathrm{Q_l})), \\ &\left.\begin{matrix} \text{cross-att}_\mathrm{c}(\mathrm{Q_c}), \mathrm{FFN_c}(\mathrm{Q_c}) \\ \text{cross-att}_\mathrm{l}(\mathrm{Q_l}), \mathrm{FFN_l}(\mathrm{Q_l}) \end{matrix}\right\}_{\times L} \end{aligned} \tag{3}$$

In this generation process, $Q_c$ and $Q_l$ are classification-aware queries and location-aware queries carefully generated by the query selection module respectively. It is worth noting that the cross-attention module and the feed-forward network module are not shared between $Q_c$ and $Q_l$. This generation process can be expressed as:

$$\mathrm{Q_c} = \mathrm{G_c}(\mathrm{F}, \mathrm{R_{box}}), \tag{4}$$

where $\mathrm{R_{box}}$ is a series of anchor boxes. We use the mini-detector module proposed in [13] to initialize those anchor boxes. $G_c$ is the task-aware query generation process which will be introduced in the next section.

### C. Query Selection Module

In the previous subsection, we have implemented the partitioning of cross-attention blocks in the decoder for disentangled feature learning. In this section, we will focus on query selection to ensure that the cross-attention of each branch can accurately focus on its specific area of interest and minimize feature conflicts. For this purpose, we specially designed a query selection module, whose structure is shown in Figure 4. Inspired by the small detectors mentioned in [13], we first generate a set of anchor boxes $\mathrm{R_{box}}$. Next, we improve the semantic alignment matching module proposed in [14]. Specifically, we use ROIAlign method to extract region-level features $\mathrm{F_R} \in \mathbb{R}^{N \times 7 \times 7 \times d}$ from the encoding feature F corresponding to the anchor box $\mathrm{R_{box}}$.

In order to capture the characteristics of the object within the anchor box, we select the most recognizable points to construct the content embedding of the object. Through RoIAlign we extract regional features $F_\mathrm{R}$. Subsequently, using convolutional neural networks and multilayer perceptrons, we further determined the precise coordinates $\mathrm{R_{SP}} \in \mathbb{R}^{N \times M \times 2}$ of these points within each region.

$$\mathrm{R_{SP}} = \mathrm{MLP}\left(\mathrm{ConvNet}\left(\mathrm{F_R}\right)\right) \tag{5}$$

In order to accurately extract the features of these points, we use bilinear interpolation method. Next, we calculated the average features of each branch and the offset of these discriminative points. These average features are then used to update the query content embedding, thereby enhancing the description of the target object. At the same time, we also apply the position encoding function PE to generate the

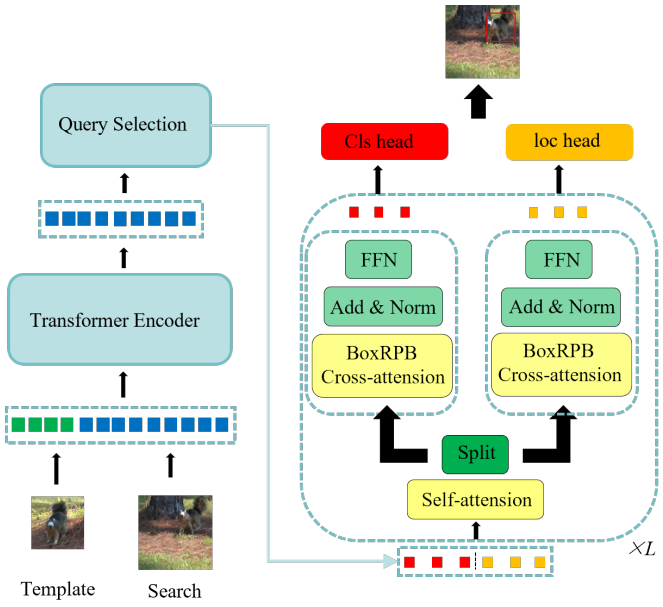


Fig. 3. Architecture of the proposed SDTrack. The key components are the query selection module and the spatially decoupled decoder. The feature misalignment problem between the two branches of the cross-attention block in the decoder is solved. Specifically, cross-attention blocks are segmented and BoxRPN terms are added, while self-attention blocks are shared to achieve information propagation.

average offset position embedding. This embedding is used to update the learnable query position embedding, thereby improving the accuracy of object localization.

### *D. Box-to-pixel Relative Position Offset*

The original cross-attention formulation usually focuses on irrelevant image regions in ordinary tracking frameworks. We speculate that this may be the reason for the poor accuracy. However, inspired by the successful application of pixel-to-pixel relative position bias in visual transformer architectures [15], we propose an innovative solution, box-to-pixel relative position bias (BoxRPB), and apply it to cross-attention calculating:

$$\mathbf{O} = \text{Softmax}(\mathbf{Q}\mathbf{K}^{\text{T}} + \mathbf{B})\mathbf{V} + \mathbf{X}, \tag{6}$$

where $\mathbf{B}$ is the relative position bias determined by the geometric relationship between boxes and pixels.

Different from the two-dimensional relative position defined by RPB, BoxRPB needs to deal with a more complex four-dimensional geometric space. In order to effectively implement BoxRPB, we do not directly calculate the bias term of the four-dimensional input. Instead, we took a more efficient approach by breaking the offset calculation into two separate terms:

$$\mathbf{B} = \text{unsqueeze}(\mathbf{B}_x, 1) + \text{unsqueeze}(\mathbf{B}_y, 2), \tag{7}$$

where $\mathbf{B}_x \in \mathbb{R}^{\mathsf{K}\times\mathsf{W}\times\mathsf{M}}$ and $\mathbf{B}_y \in \mathbb{R}^{\mathsf{K}\times\mathsf{H}\times\mathsf{M}}$ are the biases regarding $x$- axis and $y$- axis, respectively. They are computed as:

$$\mathbf{B}_x = \text{MLP}_1(\Delta\mathbf{x}_1, \Delta\mathbf{x}_2), \quad \mathbf{B}_y = \text{MLP}_2(\Delta\mathbf{y}_1, \Delta\mathbf{y}_2), \tag{8}$$

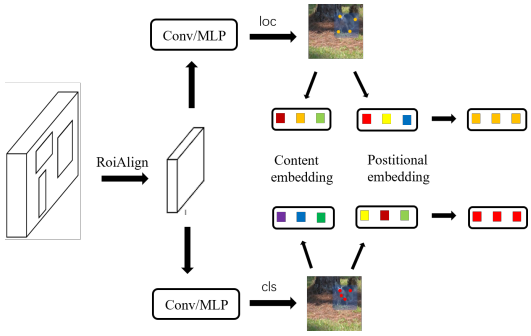


Fig. 4. Query selection module. Decouple input into classification-aware queries and location-aware queries

By decomposing, both computational FLOPs and memory consumption are significantly reduced while accuracy remains almost the same. Adding BoxRPB makes the attention focus more on objects and box boundaries, while cross-attention without BoxRPB may focus on many irrelevant areas.

### *E. Task Alignment Loss*

Previously we successfully decoupled the two branches of classification and positioning. However, when generating predictions from object queries, we discovered a significant problem: the mismatch between accurate classification and precise localization. This mismatch means that sometimes a query can produce a high-confidence classification result but a relatively low Intersection over Union (IoU) score, and vice versa. Inspired by previous studies on label assignment [6], we improve the loss function. Our goal is to achieve both high classification scores and precise localization. The

TABLE I

STATE-OF-THE-ART COMPARISON ON GOT-10K, TRACKINGNET, LASOT AND LASOT$_{ext}$. THE BEST TWO RESULTS ARE SHOWN IN RED AND BLUE FONTS, RESPECTIVELY. WE USE * TO DENOTE THAT THE RESULTS ON GOT-10K ARE OBTAINED FOLLOWING THE OFFICIAL ONE-SHOT PROTOCOL.

| Tracker | Source | GOT-10k* [16] | | | TrackingNet [17] | | | LaSOT [18] | | | LaSOT$_{ext}$ [19] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AO | $SR_{0.5}$ | $SR_{0.75}$ | AUC | $P_{Norm}$ | P | AUC | $P_{Norm}$ | P | AUC | $P_{Norm}$ | P |
| SDTrack | Ours | 75.1 | 84.4 | 72.1 | 84.4 | 89.3 | 83.9 | 72.1 | 81.7 | 79.2 | 51.5 | 62.1 | 60.1 |
| SeqTrack [20] | CVPR'23 | 74.5 | 84.3 | 71.4 | 83.9 | 88.8 | 83.6 | 71.5 | 81.1 | 77.8 | 50.5 | 61.6 | 57.5 |
| GRM [21] | CVPR'23 | 73.4 | 82.9 | 70.4 | 84.0 | 88.8 | 83.3 | 69.9 | 79.3 | 75.8 | - | - | - |
| TATrack [22] | AAAI'23 | 73.0 | 83.3 | 68.5 | 83.5 | 88.3 | 81.8 | 69.4 | 78.2 | 74.1 | - | - | - |
| MAT [23] | CVPR'23 | 64.4 | - | - | 81.9 | - | - | 67.8 | - | - | - | - | - |
| CTTrack [24] | AAAI'23 | 73.5 | 83.5 | 70.6 | 82.5 | 87.1 | 80.3 | 67.8 | 77.8 | 74.0 | - | - | - |
| AiATrack [9] | ECCV'22 | 69.6 | 80.0 | 63.2 | 82.7 | 87.8 | 80.4 | 69.0 | 79.4 | 73.8 | 47.7 | 55.6 | 55.4 |
| OSTrack [4] | ECCV'22 | 71.0 | 80.4 | 68.2 | 83.1 | 87.8 | 82.0 | 69.1 | 78.7 | 75.2 | - | - | - |
| SimTrack [2] | ECCV'22 | 68.6 | 78.9 | 62.4 | 82.3 | 86.5 | - | 69.3 | 78.5 | 74.0 | - | - | - |
| MixFormer [3] | CVPR'22 | 70.7 | 80.0 | 67.8 | 83.1 | 88.1 | 81.6 | 69.2 | 78.7 | 74.7 | - | - | - |
| SBT-B [25] | CVPR'22 | 69.9 | 80.4 | 63.6 | - | - | - | 65.9 | - | 70.0 | - | - | - |
| ToMP [10] | CVPR'22 | - | - | - | 81.2 | 86.2 | 78.6 | 67.6 | 78.0 | 72.2 | - | - | - |
| CSWinTT [26] | CVPR'22 | 69.4 | 78.9 | 65.4 | 81.9 | 86.7 | 79.5 | 66.2 | 75.2 | 70.9 | - | - | - |
| STARK [11] | ICCV'21 | 68.0 | 77.7 | 62.3 | 81.3 | 86.1 | 78.1 | 66.4 | 76.3 | 71.2 | - | - | - |
| TransT [8] | CVPR'21 | 67.1 | 76.8 | 60.9 | 81.4 | 86.7 | 80.3 | 64.9 | 73.8 | 69.0 | - | - | - |
| STMTrack [27] | CVPR'21 | 64.2 | 73.7 | 57.5 | 80.3 | 85.1 | 76.7 | 60.6 | 69.3 | 63.3 | - | - | - |
| Ocean [28] | ECCV'20 | 61.1 | 72.1 | 47.3 | - | - | - | 56.0 | 65.1 | 56.6 | - | - | - |
| SiamAttn [29] | CVPR'20 | - | - | - | 75.2 | 81.7 | - | - | - | - | - | - | - |
| DiMP [30] | ICCV'19 | 61.1 | 71.7 | 49.2 | 74.0 | 80.1 | 68.7 | 56.9 | 65.0 | 56.7 | 39.2 | 47.6 | 45.1 |
| ATOM [7] | CVPR'19 | - | - | - | 70.3 | 77.1 | 64.8 | 51.5 | 57.6 | 50.5 | 37.6 | 45.9 | 43.0 |

TABLE II

COMPARISON WITH STATE-OF-THE-ART METHODS ON TNL2K AND OTB100 BENCHMARKS IN AUC SCORE.

| Tracker | ATOM | Ocean | DiMP | TransT | Ostrack | SBT | Mixformer | OSTrack | SDTrack |
|---|---|---|---|---|---|---|---|---|---|
| TNL2K | 40.1 | 38.4 | 44.7 | 50.7 | 55.9 | - | - | 56.4 | 57.1 |
| OTB100 | 66.3 | 68.4 | 68.4 | 69.6 | - | 70.9 | 70.0 | - | 71.5 |

query allocation of training instances should meet the following rules: (i)a well-aligned query should be able to predict higher classification scores combined with accurate positioning; (ii)a misaligned query should have a low classification score and be suppressed subsequently. To achieve this, we evaluate task consistency based on a high-order combination between classification score and IoU. More specifically, we design a specific metric for calculating the consistency of each query:

$$t = s^{\alpha} \times u^{\beta} \tag{9}$$

where $s$ represents the classification score, and $u$ represents the IoU value. The parameters $\alpha$ and $\beta$ are used to control the relative influence of the two tasks in the alignment metric. Later, during the training process, we use the variable $t$ to replace the binary label of the original positive sample. This change helps the learning process dynamically give higher priority to high-quality queries. So the classification task loss function can be rewritten as:

$$\mathcal{L}_{cls} = \sum_{i=1}^{N_{\text{pos}}} |t_i - s_i|^{\gamma} BCE(s_i, t_i) + \sum_{j=1}^{N_{\text{neg}}} s_j^{\gamma} BCE(s_j, 0), \tag{10}$$

where $N_{\text{pos}}$ and $N_{\text{neg}}$ represent the number of positive samples and negative samples respectively, and $\gamma$ is the focusing parameter. To further improve the matching efficiency, we repeat the positive labels multiple times to provide richer positive supervision signals.

## IV. EXPERIMENTS

### A. Implementation Details

**Model.** We adopt the vanilla ViT-Base [31] model and initialize it with CAE [32] pre-trained weights on ImageNet as the encoder of our SDTRack. The sizes of template and search images are $128 \times 128$ pixels and $256 \times 256$ pixels respectively. All input images are segmented into $16 \times 16$ patches. We keep the number of multi-head to 8 and the attention channel to 256. The default number of queries is 300. For the architecture, we used six encoder layers and six decoder layers.

**Training.** Our experiments are conducted with an NVIDIA GeForce RTX 3090 GPUs. We use the training splits of LaSOT [18], TrackingNet [17], GOT-10k [16] and COCO [33] for training except for GOT-10k evaluation. We use the AdamW optimizer [26] and train for 300 epochs. Weight decay is set to $4 \times 10^{-4}$. The learning rates for trunk and transformer are $4 \times 10^{-5}$ and $4 \times 10^{-4}$ respectively. After 40 epochs, the learning rate drops by 10. We use a shedding rate of 0.1 for the transformer.When calculating the loss function, we use bipartite matching via the Hungarian algorithm. We repeated positive samples twice. For task alignment loss, $\alpha$, $\beta$, $\gamma$ are set to 0.25, 0.75 and 2 respectively. For box regression, to prevent supervision of low-quality tokens, we filter out and retain only the topk tokens based on predicted classification scores. We then compute the localization loss specifically for the boxes output by these selected tokens. We combine $l_1$ loss and generalized IoU loss as the training target for localization.

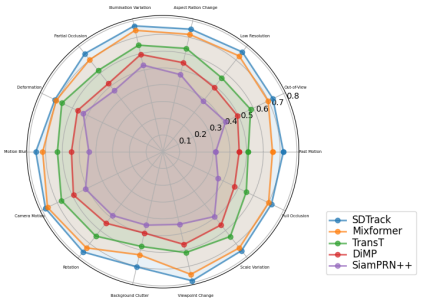


Fig. 5. AUC scores of different attributes on LaSOT [18]

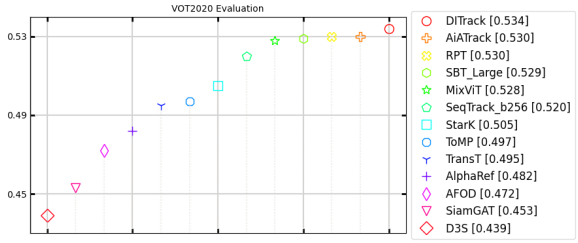


Fig. 6. EAO rank plots on VOT2020.

TABLE III

ABLATION STUDY FOR THE SDTRACK.

| | DFL | BoxRPB | TAL | AUC | Δ |
|---|---|---|---|---|---|
| SDTrack | ✓ | ✓ | ✓ | 72.1 | - |
| | | ✓ | ✓ | 69.8 | -2.3 |
| | ✓ | | ✓ | 71.3 | -0.8 |
| | ✓ | ✓ | | 70.9 | -1.2 |

### B. State-of-the-art Comparisons

To demonstrate the effectiveness of the proposed method, we evaluate and compare our SDTrack with several state-of-the-art trackers on six benchmarks.

**GOT-10k.** GOT10K is a large-scale tracking dataset containing more than 10,000 video sequences. The GOT10K benchmark proposes a protocol where the tracker is trained using only its training set. We follow this protocol to train our framework. As shown in Table 1, our tracker achieves the best AO score of 75.1%, which is 0.6% higher than the previous state-of-the-art method SeqTrack.

**TrackingNet.** TrackingNet is a large-scale short-term dataset that provides a test set of 511 video sequences. Our results are reported by an online evaluation server. Table 1 shows that our SDTrack has a success rate of 84.4% and an accuracy rate of 83.9%, exceeding all previously published trackers.

**LaSOT.** LaSOT is a large-scale long-term tracking benchmark, including 1120 sequences for training and 280 sequences for testing. From Table 1, we find that our method sets a new state-of-the-art on LaSOT, which shows that our method is also well suited for tracking scenarios with very long video sequences.

**LaSOT$_{ext}$.** LaSOT$_{ext}$ is an extended version of LaSOT and includes 150 long-term video sequences. As shown in Table 1, our method achieves good tracking results that outperform most of the compared trackers. For example, our tracker has an AUC of 51.5%, a P$_{Norm}$ score of 62.1%, and a P score of 68.9%, which are better than SeqTrack by 1.0%, 0.5%, and 2.6%, respectively.

**VOT2020.** VOT2020 contains 60 challenging sequences and uses binary segmentation masks as ground truth. We use Alpha Refine as SDTrack's post-processing network to predict segmentation masks. As shown in As shown in Figure 6, our SDTrack achieves the best results of EAO in mask evaluation.

**TNL2K and OTB100.** We evaluate our tracker on the TNL2K and OTB100 benchmarks. They include 700 and 100 video sequences respectively. These results in Table 2 show that our SDTrack achieves the best performance on TNL2K and OTB100 benchmarks.

### C. Ablations

We ablate several key components on our SDTrack using the challenging LaSOT dataset and evaluate its impact on the final results.

**The importance of DFL.** DFL refers to disentangled feature learning, which decouples the cross-attention layer. The most intuitive decoupling structure is a direct copy of the decoder, in which the classification branch and the localization branch are completely independent and do not interfere with each other. However, as shown in Table 3, this structure resulted in a performance degradation of 2.3. This is mainly because the completely decoupled design ignores the importance of information flow between the two branches. Therefore, in our design, we adopt the strategy of separating cross-attention and sharing self-attention. This not only maintains the flow of information between different branch queries, but also reduces the introduction of additional parameters, thereby achieving more efficient and accurate model design.

**The importance of BoxRPB.** As shown in the Table 3. When we remove the BoxRPB term from cross-attention, the performance gain drops by 0.8. We speculate that while the original cross-attention usually focuses on irrelevant areas within the image, the BoxRPB term enables the decomposed classification branch and localization branch to focus on their respective object areas of interest, improving the accuracy of localization.

**The importance of TAL.** As we analyzed above, the misalignment between accurate classification and precise positioning will hinder the performance of the tracker, and

TABLE IV

ANALYSIS OF DIFFERENT HYPER-PARAMETERS.

| AUC | 71.5 | 70.6 | **72.1** | 70.1 | 69.9 |
|---|---|---|---|---|---|
| $\alpha$ | 0.25 | 0.25 | 0.25 | 0.5 | 0.5 |
| $\beta$ | 0.50 | 0.50 | 0.75 | 0.75 | 1.0 |
| $\gamma$ | 1.0 | 2.0 | 2.0 | 2.0 | 3.0 |

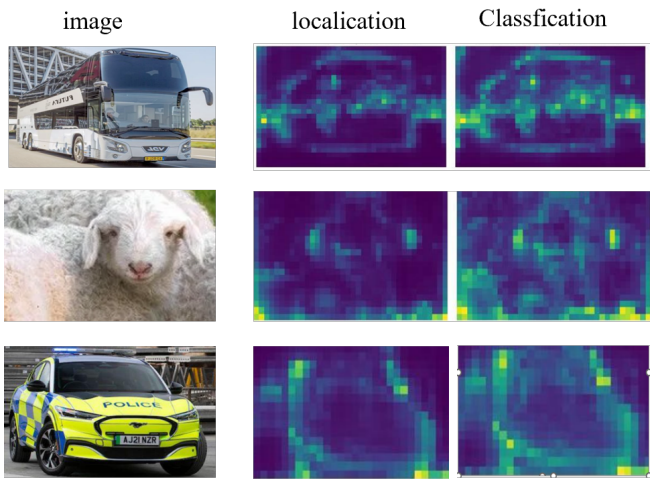


Fig. 7. Visualization of cross-attention maps for the classification branch and localization branch of our SDTrack. The high activation area differs for the two branches.

the data in Table 3 also confirms our view. Compared with the tracker using TAL, the performance of not using TAL dropped by 1.2, so using TAL to reduce the misalignment between the two tasks is necessary to improve the performance of the tracker.

**On hyper-parameters.** We investigate the performance using different values of $\alpha$ and $\beta$ for TAL, which control the influence of classification confidence and localization precision on anchor alignment metric via $t$. From Table 4, we can know that the best effect is achieved when $\alpha = 0.25$, $\beta = 0.75$ and $\gamma = 2$ are used, which encourages the network to dynamically focus on high-quality queries from the perspective of joint optimization. Therefore, it enables our tracker to more accurately locate specific targets.

### *D. Visualizations*

Figure 7 shows the cross-attention map for classification and localization branches. Spatial attention maps show that classification and localization tasks have different regions of high activation, supporting our hypothesis that features from different object locations contribute differently to these tasks. The localization branch tends to have higher activation on object edges, while the classification branch pays more attention to the overall object, especially the salient areas. By decoupling the two branches, each branch can capture its unique information more flexibly.

## V. CONCLUSION

Our study proposes a spatially decoupled tracker model that successfully addresses the problem of feature and prediction misalignment in classification and localization tasks. In detail, we implement cross-attention block segmentation in the decoder and incorporate BoxRPB terms in the cross-attention mechanism. This design enables each branch to focus on its own area of concern while sharing self-attention blocks. In order to further reduce the misalignment between high classification confidence and precise localization, we also specifically introduce a task alignment loss. Through our experimental verification, the effectiveness of this method is fully demonstrated. In the future, we will continue to work on exploring more complex transformer cross-attention structures in order to further achieve information decoupling.

## REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Inf. Process. Syst.*, vol. 30, 2017.

[2] B. Chen, P. Li, L. Bai, L. Qiao, Q. Shen, B. Li, W. Gan, W. Wu, and W. Ouyang, "Backbone is all your need: A simplified architecture for visual object tracking," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2022, pp. 375–392.

[3] Y. Cui, C. Jiang, L. Wang, and G. Wu, "Mixformer: End-to-end tracking with iterative mixed attention," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2022, pp. 13 608–13 618.

[4] B. Ye, H. Chang, B. Ma, S. Shan, and X. Chen, "Joint feature learning and relation modeling for tracking: A one-stream framework," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2022, pp. 341–357.

[5] Q. Wei, G. Zeng, and B. Zeng, "Efficient training for visual tracking with deformable transformer," *arXiv preprint arXiv:2309.02676*, 2023.

[6] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "Tood: Task-aligned one-stage object detection," in *Proc. IEEE Int. Conf. Comp. Vis.* IEEE Computer Society, 2021, pp. 3490–3499.

[7] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019, pp. 4660–4669.

[8] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2021, pp. 8126–8135.

[9] S. Gao, C. Zhou, C. Ma, X. Wang, and J. Yuan, "Aiatrack: Attention in attention for transformer visual tracking," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2022, pp. 146–164.

[10] C. Mayer, M. Danelljan, G. Bhat, M. Paul, D. P. Paudel, F. Yu, and L. Van Gool, "Transforming model prediction for tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2022, pp. 8731–8740.

[11] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2021, pp. 10 448–10 457.

[12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2020, pp. 213–229.

[13] L. He and S. Todorovic, "Destr: Object detection with split transformer," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2022, pp. 9377–9386.

[14] G. Zhang, Z. Luo, Y. Yu, K. Cui, and S. Lu, "Accelerating detr convergence via semantic-aligned matching," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2022, pp. 949–958.

[15] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2022, pp. 12 009–12 019.

[16] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, 2019.

[17] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. Eur. Conf. Comp. Vis.*, 2018, pp. 300–317.

[18] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019, pp. 5374–5383.

[19] H. Fan, H. Bai, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, Harshit, M. Huang, J. Liu *et al.*, "Lasot: A high-quality large-scale single object tracking benchmark," *Int. J. Comput. Vision*, vol. 129, pp. 439–461, 2021.

[20] X. Chen, H. Peng, D. Wang, H. Lu, and H. Hu, "Seqtrack: Sequence to sequence learning for visual object tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2023, pp. 14 572–14 581.

[21] S. Gao, C. Zhou, and J. Zhang, "Generalized relation modeling for transformer tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2023, pp. 18 686–18 695.

[22] K. He, C. Zhang, S. Xie, Z. Li, and Z. Wang, "Target-aware tracking with long-term context attention," in *Proc. Conf. AAAI*, vol. 37, no. 1, 2023, pp. 773–780.
[23] H. Zhao, D. Wang, and H. Lu, "Representation learning for visual object tracking by masked appearance transfer," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2023, pp. 18 696–18 705.
[24] Z. Song, R. Luo, J. Yu, Y.-P. P. Chen, and W. Yang, "Compact transformer tracker with correlative masked modeling," in *Proc. Conf. AAAI*, vol. 37, no. 2, 2023, pp. 2321–2329.
[25] F. Xie, C. Wang, G. Wang, Y. Cao, W. Yang, and W. Zeng, "Correlation-aware deep tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2022, pp. 8751–8760.
[26] Z. Song, J. Yu, Y.-P. P. Chen, and W. Yang, "Transformer tracking with cyclic shifting window attention," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2022, pp. 8791–8800.
[27] Z. Fu, Q. Liu, Z. Fu, and Y. Wang, "Stmtrack: Template-free visual tracking with space-time memory networks," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2021, pp. 13 774–13 783.
[28] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2020, pp. 771–787.
[29] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable siamese attention networks for visual object tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020, pp. 6728–6737.
[30] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019, pp. 6182–6191.
[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *Proc. Int. Conf. Learn. Representations*, 2020.
[32] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, and J. Wang, "Context autoencoder for self-supervised representation learning," *Int. J. Comput. Vision*, vol. 132, no. 1, pp. 208–223, 2024.
[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2014, pp. 740–755.